

Alma Mater Studiorum – Università di Bologna

DOTTORATO DI RICERCA IN

**Meccanica e Scienze Avanzate dell'Ingegneria
INGEGNERIA DELLE MACCHINE E DEI SISTEMI ENERGETICI**

Ciclo XXVIII

Settore Concorsuale di afferenza: 09/C1

Settore Scientifico disciplinare: ING-IND/08

**DEVELOPMENT AND ASSESSMENT OF
LARGE EDDY SIMULATION METHODOLOGY
FOR INTERNAL COMBUSTION ENGINES**

Presentata da: Cristian Catellani

Coordinatore Dottorato

Relatore

Prof. Vincenzo Parenti Castelli

Prof. Gian Marco Bianchi

Esame finale anno 2016

***"Essentially, all models are wrong,
but some are useful."***

George E. P. Box

To my Parents

Abstract

Large Eddy Simulation (LES) represents nowadays one of the most promising techniques for the evaluation of the dynamics and evolution of turbulent structures characterizing Internal Combustion Engines (ICE).

The demand for a high level of resolution accuracy as well as the need to evaluate different scenarios and system configurations lead to considerable computational and economic costs for both the hardware infrastructure and the licensing fees of commercial codes.

In such context, the present Doctoral project has the objective to define the most suitable numerical methodology to perform LES analysis of ICE flows and to implement such methodology in an efficient, accurate and robust CFD code, based on open-source components.

An evaluation of freely available CFD codes has led to the choice of the open-source CFD package *OpenFOAM*[®] as the most suited code for the project's objective.

The LES modeling of interest for ICE applications has been then studied and three Sub-grid scale models particularly suited for such flows have been implemented and assessed into *OpenFOAM*[®]. Moreover, Python scripts have been developed in order to automate and speed-up both pre-processing and post-processing phases.

The CFD methodology has been then applied to a real world ICE systems such as a stationary flow bench, for which prior RANS simulations had shown some predictive deficiencies. The quality of the analyses has been assessed through specific LES quality estimators and the computational results have been validated against measurements, showing pretty

good agreement.

Finally, LES simulations have allowed the accurate investigation of the flow bench fluid-dynamic behavior and, thanks to the insights gained, an alternative RANS approach based on the Reynolds Stress Tensor Modeling has been proposed and tested in order to alleviate the aforementioned predictive deficiencies.

Keywords: Computational Fluid Dynamics, CFD, Large Eddy Simulation, LES, Internal Combustion Engine, ICE, Sub-grid Scale model, SGS, Flow Test Bench.

Sommario

La Large Eddy Simulation (LES) rappresenta oggi una delle più promettenti metodologie computazionali per lo studio della dinamica e dell'evoluzione dei flussi turbolenti caratterizzanti i motori a combustione interna.

Tale modellistica presenta superiori capacità di risoluzione rispetto alla tradizionale metodologia RANS, ma è anche caratterizzata da maggiori oneri computazionali nonché economici per l'infrastruttura di calcolo e per i costi di licensing dei codici commerciali.

Il presente Dottorato di Ricerca ha quindi l'obiettivo di definire la metodologia numerica LES più idonea per lo studio di flussi motoristici, implementando poi tale modellistica in un codice di calcolo efficiente, robusto, accurato e basato su componenti open-source.

La disamina delle risorse software liberamente fruibili ha permesso di individuare nel pacchetto open-source *OpenFOAM*[®] le librerie e gli applicativi più idonei allo scopo.

Si è quindi proceduto allo studio e alla definizione della modellistica LES per flussi motoristici e all'implementazione in *OpenFOAM*[®] di modelli SGS idonei a tali applicazioni, validando i risultati attraverso l'impiego di differenti test-cases documentati in letteratura. Un pacchetto di script Python appositamente realizzato ha inoltre permesso di automatizzare le fasi di pre-processing e post-processing, accelerando l'esecuzione delle analisi.

La metodologia di calcolo è stata quindi applicata ad un caso motoristico reale quale un banco di flussaggio stazionario, per il quale precedenti simulazioni RANS avevano mostrato scarsa accuratezza predittiva.

La qualità delle analisi è stata misurata grazie a specifici indicatori, mentre il confronto tra i risultati computazionali e i dati sperimentali ha dimostrato le buone capacità predittiva del codice ottenuto.

Le simulazioni LES hanno permesso di studiare dettagliatamente la fluidodinamica del sistema. La disamina dei campi di moto ottenuti ha consentito infine di individuare potenziali cause della scarsa accuratezza delle originarie simulazioni RANS e di proporre, per esse, un differente approccio modellistico al fine di alleviare tale deficit predittivo.

Keywords: Computational Fluid Dynamics, CFD, Large Eddy Simulation, LES, Internal Combustion Engine, ICE, Sub-grid Scale model, SGS, Flow Test Bench.

Contents

1	Introduction	3
1.1	Motivation	3
1.2	Research objectives	4
1.3	Thesis structure	6
2	LES modeling of ICE	7
2.1	Computational modeling of ICEs	7
2.2	Governing equations	10
2.2.1	Basic assumptions and generalized transport equation	10
2.2.2	Governing equations for compressible flows	10
2.2.3	Governing equations for incompressible flows	14
2.3	LES modeling	15
2.3.1	LES filtering operations	15
2.3.2	The filtered Navier-Stokes equations	18
2.3.3	Sub-Grid Scale modeling	20
2.4	Implemented sub-grid scale models	22
2.4.1	The dynamic Smagorinsky SGS model	22
2.4.2	The WALE SGS model	23
2.4.3	The dynamic WALE SGS model	24
2.4.4	The Sigma SGS model	25
2.5	The Finite Volume Method	26
2.5.1	Numerical methods for CFD simulations	26
2.5.2	The Finite Volume discretization process	26
2.5.3	Time discetization	28

2.5.4	Space discretization	29
2.5.5	Solution of linear equations systems	33
2.5.6	Pressure-velocity coupling	34
2.5.7	A few words about <i>OpenFOAM</i> [®]	37
2.6	Numerical dissipation and dispersion	38
3	LES Quality Assessment	43
3.1	General remarks on LES analysis quality assessment	43
3.2	LES quality estimators	44
3.2.1	Fraction of resolved kinetic energy - M	44
3.2.2	Length Scale Resolution - LSR	45
3.2.3	Viscosity ratio - ν_{SGS}/ν	46
3.2.4	Sub-grid Scale Activity - s	46
3.2.5	Two grid estimator - LES _{IQk}	47
4	Basic Test Cases	49
4.1	Dellenback abrupt expansion	49
4.1.1	Test case and computational setup	49
4.1.2	Results - Velocity profiles	51
4.1.3	Results - Quality assessment	61
4.1.4	Results - Energy spectra	62
4.1.5	Results - Pressure SPD	62
4.2	Thobois poppet valve	68
4.2.1	Test case and computational setup	68
4.2.2	Results - Velocity profiles	71
4.2.3	Results - Quality assessment	72
4.2.4	Results - Energy spectra	73
4.3	Computational aspects	83
4.4	Closure	84
5	Virtual Flow Bench	87
5.1	Flow bench general notes	87
5.2	Eddy-viscosity RANS approach	89
5.2.1	Case setup	89
5.2.2	Impulse swirl meter modeling	92
5.2.3	Results	94
5.3	LES approach	100
5.3.1	Case setup	100
5.3.2	Quality assessment	102
5.3.3	Results	107
5.4	RSTM RANS approach	120

5.5	Summary of results	123
5.6	Closure	124
6	Summary and Outlook	127
6.1	Research summary	127
6.2	Ongoing activities	130
6.3	Future work	131
6.4	Scientific production	132
	Appendices	133
A	Implemented SGS models source code	135
A.0.1	Dynamic WALE SGS model	135
A.0.2	Sigma SGS model	143
B	Mass Flow Rate and Swirl Torque evaluation methodology	151
B.0.3	Mass Flow Rate and Swirl Torque evaluation	151
B.0.4	Post-processing script	155
	Bibliography	159

List of Figures

2.1	Turbulent kinetic energy decay.	9
2.2	Filtering of a randomly fluctuating variable f	16
2.3	Control volume topology.	28
2.4	Flowchart of the PISO algorithm.	36
2.5	Taylor vortex test setup.	39
2.6	Mesh influence - Relative pressure profile after 1 c.t.	41
2.7	Mesh influence - Relative pressure profile after 5 c.t.	41
2.8	Co influence - Relative pressure profile after 1 c.t.	41
2.9	Co influence - Relative pressure profile after 5 c.t.	41
2.10	Numerical dissipation as a function of the c.t.	41
2.11	Numerical dispersion as a function of the c.t.	41
3.1	Number of published articles concerning LES of ICE.	44
4.1	Geometry of the Dellenback test case and probe locations.	50
4.2	Mean axial velocity profiles - coarse mesh	53
4.3	Mean tangential velocity profiles - coarse mesh	54
4.4	RMS of axial velocity fluctuations - coarse mesh	55
4.5	RMS of tangential velocity fluctuations - coarse mesh	56
4.6	Mean velocity profiles for different grid densities	57
4.7	RMS of velocity fluctuations for different grid densities	58
4.8	Mean velocity profiles for different grid element types	59
4.9	RMS of velocity fluctuations for different grid element types	60
4.10	M parameter for different grid densities	63

List of Figures

4.11 LSR parameter for different grid densities	63
4.12 ν_{SGS}/ν parameter for different grid densities	64
4.13 LES_{IQk} parameter comparison maps	64
4.14 LES_{IQk} parameter at different locations	65
4.15 T.k.e. spectra for different grid densities	65
4.16 T.k.e. spectra for different grid element type	66
4.17 T.k.e. spectra for different SGS models	66
4.18 Pressure SPD for different grid densities	67
4.19 Pressure SPD for different grid element type	67
4.20 Geometry of the Thobois test case and probe locations.	69
4.21 Coarse hexahedral grids for the Thobois test case.	70
4.22 Mean and RMS velocities for different grid densities	74
4.23 Mean and RMS velocities for different grid densities	75
4.24 Mean and RMS velocities for different grid element types	76
4.25 Mean and RMS velocities for different grid alignments	77
4.26 M parameter for different grid densities	78
4.27 LSR parameter for different grid densities	79
4.28 ν_{SGS}/ν parameter for different grid densities	80
4.29 T.k.e. spectra for different grid densities	81
4.30 T.k.e. spectra for different grid element type	81
4.31 T.k.e. spectra for different SGS models	82
4.32 T.k.e. spectra for different locations	82
4.33 <i>OpenFOAM</i> [®] scalability tests	84
5.1 Generic induced steady flow bench scheme	88
5.2 Computational domain geometry.	90
5.3 Flow bench computational grid.	91
5.4 Schematic of steady-flow impulse swirl meter	92
5.5 Porous region location.	93
5.6 Post-processing topology used for the Virtual Flow Bench.	93
5.7 Details of the flow in the valve region	95
5.8 Mass flow rate as a function of valve lift	96
5.9 Swirl torque as a function of valve lift	96
5.10 Visualization of the porous region flow straightening	97
5.11 Velocity magnitude map on the valves plane	98
5.12 Details of the velocity magnitude close to valve ports.	99
5.13 Details of the turbulent kinetic energy close to valve ports.	99
5.14 Grid comparison for the 5 mm lift. RANS vs LES	101
5.15 Details of the mean M parameter close to valve ports	103
5.16 Instantaneous y^+ map for the 5 mm valve lift.	104

5.17 M parameter map on the valves plane. Valve lift 5 mm. . . .	105
5.18 Viscosity ratio map on the valves plane. Valve lift 5 mm. . .	105
5.19 Details of the mean SGS t.k.e. close to valve ports	106
5.20 Details of the mean visc. ratio close to valve ports	106
5.21 Mass flow rate for the 10 mm valve lift	107
5.22 Mass flow rate as a function of valve lift	110
5.23 Swirl torque as a function of valve lift	110
5.24 Instantaneous velocity mag. map for the 5 mm valve lift. . .	111
5.25 Instantaneous absolute pressure map for the 5 mm valve lift.	111
5.26 Instantaneous SGS t.k.e. map for the 5 mm valve lift.	112
5.27 Instantaneous viscosity ratio map for the 5 mm valve lift. . .	112
5.28 Spectra probes locations.	113
5.29 T.k.e. spectra for different locations	113
5.30 Pressure SPD for different locations	114
5.31 Mass flow rate SPD	114
5.32 Post-processing topology for planar streamlines.	116
5.33 Sketch of vortexes development inside the cylinder.	117
5.34 Planar streamlines for 2 mm valve lift.	118
5.35 Planar streamlines for 5 mm valve lift.	119
5.36 Mass flow rate as a function of valve lift - RSTM	121
5.37 Swirl torque as a function of valve lift - RSTM	121
5.38 Planar streamlines for 5 mm valve lift - RSTM.	122
 B.1 Swirl torque measurement section location.	 152
B.2 Swirl torque computation.	153

List of Tables

4.1	Main geometric parameters of the Thobois test case.	70
4.2	Computational requirements of the different SGS models. . .	83
5.1	Engine parameters.	89
5.2	Virtual flow bench boundary conditions (RANS).	91
5.3	Virtual flow bench boundary conditions (LES).	102
5.4	Steady flow bench mass flow rates.	123
5.5	Steady flow bench swirl torques.	123

Nomenclature

Abbreviations

<i>ACEA</i>	Association des Constructeurs Européens d'Automobiles
<i>FD</i>	Finite Difference
<i>FE</i>	Finite Element
<i>FV</i>	Finite Volume
<i>GDI</i>	Gasoline Direct Injection
<i>HCCI</i>	Homogeneous Charge Compression Ignition
<i>ICE</i>	Internal Combustion Engine
<i>LBA</i>	Lattice Boltzmann Automata
<i>LDA</i>	Laser-Doppler Anemometry
<i>LDV</i>	Laser Doppler Velocimetry
<i>LES</i>	Large Eddy Simulation
<i>LHS</i>	Left Hand Side
<i>LSR</i>	Length Scale Resolution
<i>NS</i>	Navier-Stokes
<i>PIV</i>	Particle Image Velocimetry

List of Tables

PM	Particle Methods
$RANS$	Reynolds Averaged Navier Stokes
RHS	Right Hand Side
$RSTM$	Reynolds Stress Transport Model
SGS	Sub-Grid Scale
SM	Spectral Methods
SPD	Spectral Power Density
SVS	Shear and Vortex Sensor
$WALE$	Wall-Adapting Local Eddy-viscosity

Greek letters

Δ	Filter Width
η	Kolmogorov Microscale Length
μ	(Molecular) Dynamic Viscosity
μ_{SGS}	Sub-grid Eddy (Dynamic) Viscosity
ν	(Molecular) Kinematic Viscosity
ν_{SGS}	Sub-grid Eddy (Kinematic) Viscosity
ρ	Density
σ_i	Velocity Gradient Tensor Singular Values
τ_{ij}^{SGS}	Sub-grid Stress Tensor
ε_{SGS}	Sub-grid Turbulent Dissipation

Latin letters

Co	Courant Number
E	Total Energy
e	Internal Energy
g_{ij}	Velocity Gradient Tensor
h	Static Enthalpy

K	Kinetic Energy
k	Turbulent Kinetic Energy
k_{num}	Numerical Turbulent Kinetic Energy
k_{res}	Resolved Turbulent Kinetic Energy
k_{SGS}	Sub-grid Turbulent Kinetic Energy
k_{tot}	Total Turbulent Kinetic Energy
l_{DI}	Inertial Subrange Lower Limit
L_{ij}	Leonard Stress Tensor
p	Static Pressure
Re	Reynolds Number
S	Swirl Number
S_{ij}	Resolved Rate-of-strain Tensor
T	Static Temperature
u	Velocity

CHAPTER 1

Introduction

1.1 Motivation

Internal Combustion Engines (ICE) play nowadays a major role between several other power generation technologies. Whenever low to medium power is requested or in case of mobile scenarios, ICEs are advantageous, since they provide high power to weight ratios, simplicity, compactness, high reliability and low production costs. It is therefore not surprising that the ICE is the most widely used propulsion system in the automotive industry: accordingly to ACEA [1], in 2014 97.3% of EU passenger cars were adopting either a gasoline or a Diesel engine.

In the last decade most of the automotive industry R&D efforts have been devoted to increase engines fuel efficiency and reduce pollutant emissions, spending more than 41 Billion € in EU only [1]. Since EU directives introduction in 1995, the improvements have been impressive: the average CO₂ output of passenger cars has fallen by 20%, while polluting emissions dropped by more than 86% for NO_x and 95% for PM [1].

Nevertheless, the increasing severity of the upcoming regulations pose huge challenges to manufacturers and push for continuous advances in engines performances.

In order to comply with such regulations, reducing fuel consumption and limiting pollutant emissions, improved knowledge of all the physical processes occurring in the engine is mandatory. Among these, the most important ones are those related to the gas exchange and combustion stages which can be studied either through experiments or by simulations.

It is therefore not surprising the popularity that has gained over the last decades the simulation through Computational Fluid Dynamics (CFD). This technique allows in fact to reach a level of detail that is difficult to obtain experimentally and to evaluate the performance of the engine components at an early design stage, thus reducing the prototyping effort with tremendous benefits in terms of cost savings.

In ICE flows simulations the most widely used approach is the Reynolds Averaged Navier-Stokes (RANS) method. In this approach the Navier-Stokes equations are time averaged, therefore losing details on the flow turbulent fluctuations. This leads to one of the major issues with RANS modeling, i.e. its intrinsic inability to predict the unsteady flow structures characterizing ICE flows which strongly affect complex phenomena like fuel-air mixing and cycle-to-cycle engine variability.

With the advent of new engine concepts like GDI, HCCI etc. and with the always more stringent legislation regarding pollutant emission limitations and engine efficiency, the adoption of more accurate predictive models is not an option anymore and here is where Large Eddy Simulation (LES) can make the difference.

In the LES methodology the Navier-Stokes equations are filtered in space, directly resolving the largest scales of motion, while the smaller scales are taken into account through dedicated models. Thanks to this approach, LES provides a more physical representation of the turbulent structures than RANS, being capable of reproducing most of the unsteady flow features characterizing turbulent flows.

1.2 Research objectives

While LES represents nowadays one of the most promising techniques for the evaluation of the dynamics and evolution of turbulent structures characterizing ICE flows, the demand for an increased level of resolution accuracy, as well as the need to evaluate different scenarios and system configurations, lead to considerable computational and economic costs for both the hardware infrastructure and the licensing fees of commercial codes.

In such context, the present Doctoral project has the objective to define the most suitable numerical methodology to perform LES analysis of

ICE flows and to implement such methodology in an efficient, accurate and robust CFD code, based on open-source components. This will allow to:

- Avoid licensing fees, thus reducing simulation costs.
- Extend and/or modify the code as needed.
- Promote joint development and collaboration with other research groups.

In particular, the work here presented aims primarily at increasing the accuracy resolution of the intake flow field through stationary rig testing. A better understanding of it allows engine developers to design an intake geometry for the operating engine that provides a flow field (charge rotation and turbulence) at the Start of Ignition (SOI) optimized for the subsequent combustion process.

The research has initially dealt with the review of the literature of interest for the project, with particular emphasis on computational modeling of ICE flows, LES methods and LES analysis quality assessment. Then, an evaluation of freely available CFD codes has led to the choice of the open-source CFD package *OpenFOAM*[®] as the most suited code for the project's objective.

In the subsequent phase, the numerical models of interest have been defined, implemented and assessed into *OpenFOAM*[®] so obtaining a computational tool capable of effectively tackling the flows of interest. Moreover, Python scripts have been developed in order to automate and speed-up both pre-processing and post-processing phases.

In order to assess the resolution accuracy provided by the simulations, different classes of LES quality parameters have been studied. For some of these estimators, the ones most suitable for ICE flows applications, specific Python scripts have been developed in order to automate quality evaluation during post-processing.

The CFD code has been then applied to a real world ICE system such as a stationary flow bench. The engine's intake flow details are studied and compared with RANS results, highlighting some predictive deficiencies of the latter. Furthermore, the quality of the analyses has been evaluated through LES quality estimators and a comparison of the computational results against the experimental data has been performed.

Finally, thanks to the insights gained through the LES study, an alternative RANS approach based on RSTM modeling is proposed in order to alleviate the aforementioned predictive deficiency.

1.3 Thesis structure

The thesis is structured as follows:

- In Chapter 2 some basic considerations regarding numerical modeling of ICEs are introduced. Then the governing equations of fluid motion are presented along with the LES modeling approach to such flows adopted in the present work. A section is dedicated to the discussion of the Sub-Grid Scales (SGS) models used for the analyses. Such models, namely the WALE, the dynamic WALE and the Sigma models, have been implemented into the CFD solver of choice (*OpenFOAM*[®]). Basic details of the Finite Volume Method (FVM) are then summarized, since the FVM is the numerical framework under which *OpenFOAM*[®] operates. Finally, the chapter is closed with a few notes on numerical dissipation and dispersion issues, whose effects could potentially jeopardize LES simulations accuracy.
- Chapter 3 opens with general remarks on LES analysis assessment and continues presenting the details of single-grid and multi-grid quality estimators adopted in this work. Indications and limitations of their usage are also provided.
- In Chapter 4 the implemented SGS models are employed in two different test cases of interest for ICE flows: an abrupt expansion of a swirled flow and the flow past a poppet valve in a stationary engine test bench. In both cases the results are validated using experimental data and the LES resolution level is verified through the quality estimators presented in Chapter 3. The chapter ends with a few notes related to the computational cost of the analyses and with the results of scalability tests performed with the code.
- Chapter 5 presents the application of the LES methodology to a real world ICE system, in particular to a steady flow bench for the performance evaluation of a Diesel engine head. Firstly, RANS results are presented and some predictive deficiencies are highlighted. Then LES analyses are performed in order to provide accurate flow predictions and to tackle the causes of the aforementioned RANS issues. The quality of LES simulations is evaluated and the results are validated against experimental data. The chapter closes with some final remarks and advice related to stationary flow bench simulations that could be usefully employed in industrial development and optimization processes.

CHAPTER 2

LES modeling of ICE

2.1 Computational modeling of ICEs

The first attempts of ICE modeling can be dated back to 1950's, when the available computational capabilities allowed only for the calculation of simple mathematical formulae. An example of this is the Wiebe model [2], a zero-dimensional empirical model used to predict engine burn fraction and burn rate. A further example of this kind is the Woschni model [3] for engine heat transfer computations. All these studies demonstrated that the combination of empirically based engine models with optimally tuned model parameters was able to match reasonably well pressure traces of engine experimental measurements.

A natural evolution of these methods has been the upgrade from 0-D models to 1-D models encompassing engine components other than the combustion chamber. For instance, these system-level codes allowed to extend the computational domain to the engine runners and valve ports, so that the aerodynamics of intake and exhaust systems on the engine performance could be taken into account. Despite still relying to a certain degree on empirical correlations, the 1-D approach has had great success in industry where it is nowadays commonly employed as development tool.

Several commercial and open-source 1-D codes have been developed during the years and today *GT-Power*[®] is probably the most widely adopted one.

Early 1970's have seen the advent of 3-D CFD ICE (in-cylinder) modeling, even if for a decade this hasn't been generally applied to engine development. The reason for this is twofold: firstly, computational resources were still limited, secondly, engine CFD codes were still not widely available. These limitations have been overcome in 1980's, when a group in Los Alamos National Laboratory developed and released the open-source code *KIVA*[®] [4] that integrated several engine CFD modeling components, such as mesh motion, compressible flows, spray dynamics and combustion chemistry. Since then, many other codes followed. Some of them, such as *AVL Fire*[®], *Ricardo Vectis*[®] or *CS Converge*[®], have been specifically developed for tackling engine flows. Others, such as the general-purpose code *ANSYS Fluent*[®], have been equipped with modules that allow for ICE flows computations. Unfortunately no open-source code is today specifically developed for engine applications, apart from *KIVA*[®] which is aging and is becoming less and less capable of adapting to the recent engine development challenges and needs.

Entering the new millennium, ICE simulation has faced another major paradigm shift, proceeding from the RANS modeling (the almost universally adopted approach to turbulent flows since then), towards scale resolving methods, such as LES. As mentioned in Section 1.1, this has been the result of the industry need for greater accuracy and more detailed and realistic results.

Simulations must then resolve the most important scales of motion within the cylinder. From this perspective and with reference to Figure 2.1, the turbulence modeling approaches can be sorted in three main groups:

- **RANS (Reynolds Averaged Navier-Stokes):** the Navier-Stokes equations are averaged in time (Reynolds-averaged). Extra terms appear in the averaged flow equations due to the interactions between the fluctuating terms. Such extra terms are modeled with classical turbulence models such as the well-known $k - \epsilon$. This approach allows resolving the mean flow and models the effects of turbulence on the mean flow properties. No fluctuating term/contribution is directly resolved so that with this approach the turbulent spectrum is completely modeled.
- **LES (Large Eddy Simulation):** the unsteady Navier-Stokes equations are filtered in space prior to the computations, passing the largest eddies and rejecting the smaller ones. The effects of the unresolved ed-

dies on the resolved flow field are managed by the Sub-Grid Scale (SGS) models. The filter width determines the length scale cutoff in the turbulent energy spectrum. This approach has resolution capabilities that lay in between RANS and DNS.

- DNS (Direct Numerical Simulation): the unsteady Navier-Stokes equations are solved as they are. The grid employed has to be sufficiently fine to resolve up to the Kolmogorov length scales at which energy dissipation occurs. Time steps have to be sufficiently small to properly resolve the evolution of the fastest fluctuations within the flow. With this approach no turbulence modeling is performed and the turbulent energy spectrum is fully resolved.

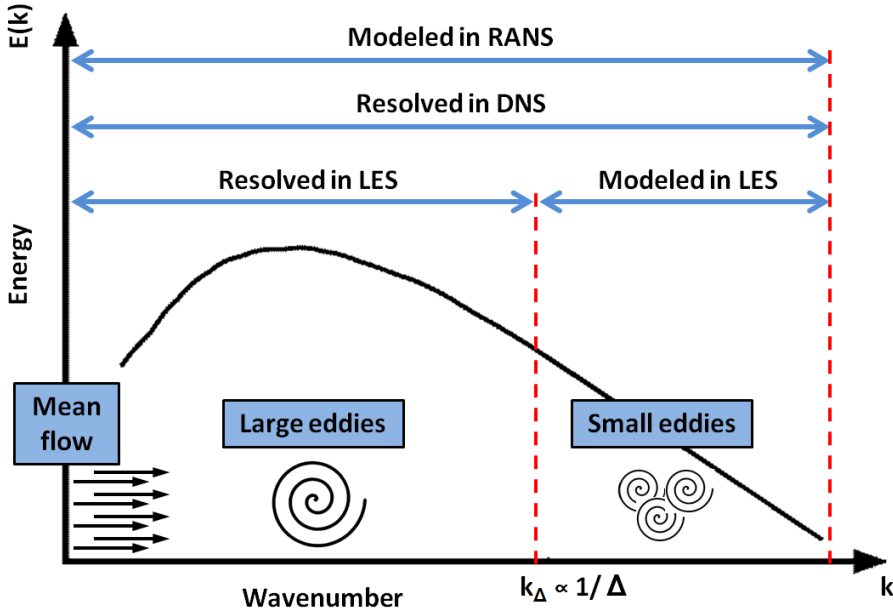


Figure 2.1: *Turbulent kinetic energy decay.*

Amongst the scale resolving approaches to turbulence, LES, and eventually some of its hybrid variants, seems promising for ICE simulations in terms of both results fidelity and computational costs: while RANS is computationally cheap but intrinsically incapable of resolving any turbulent scale, DNS directly resolves all scales of motion but has prohibitive computational costs [5] and its application to engine flows remains utopic for decades to come.

2.2 Governing equations

2.2.1 Basic assumptions and generalized transport equation

ICE flows object of the present research can be studied under the theoretical framework of continuum fluid mechanics [6]: length and time scales involved in the analyses are in fact significantly larger than the scales of the discrete structure of matter. Under this hypothesis, the fluid is assumed to be completely filling the space it occupies and to be continuously distributed in such space, i.e. with properties varying continuously from one point to another.

The equations that follow have been derived under such assumptions and considering only gaseous monophasic non-reactive fluids. Furthermore, the set of governing equations are firstly presented in their compressible formulation, since for ICE flows compressibility effects are generally not-negligible. However, for the basic test cases of Chapter 4 an incompressible formulation has been conveniently adopted, and this latter formulation will be presented too.

From the Reynolds' transport theorem [7], the sets of equations of interest will be derived through a generalized conservation equation that can be written as:

$$\frac{\partial}{\partial t} \int_V U dV + \oint_{\Omega} \mathbf{F} \cdot \mathbf{n} dA = \int_V Q_v dV + \oint_{\Omega} \mathbf{Q}_{\Omega} \cdot \mathbf{n} dA \quad (2.1)$$

where U is a conserved scalar quantity on the control volume V , delimited by the surface Ω and F is its flux through such surface. Q_v and Q_{Ω} are the source terms, respectively of the volume V and surface Ω . In case of a vector quantity \mathbf{U} , the conservation equation becomes:

$$\frac{\partial}{\partial t} \int_V \mathbf{U} dV + \oint_{\Omega} \mathbf{F} \cdot \mathbf{n} dA = \int_V \mathbf{Q}_v dV + \oint_{\Omega} \mathbf{Q}_{\Omega} \cdot \mathbf{n} dA \quad (2.2)$$

where the previous vector flux term and surface source term have been substituted by the corresponding tensors.

2.2.2 Governing equations for compressible flows

From the general Equations (2.1) and (2.2), the conservation equations for mass, momentum and energy can be derived [8]. These equations will be written per unit volume, so that for the mass the conserved quantity U will

be the density ρ , for the momentum it will be the volumetric momentum ρu , where u is the fluid velocity and for the energy it will be the specific total energy E . The compressible set of Navier-Stokes equations for compressible fluid flow consists then in:

- **Mass balance (or Continuity equation)**

States the conservation of mass. No mass sources are considered and no diffusive fluxes exist for the mass transport, thus only the convective flux is present in the equation:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{U}) = 0 \quad (2.3)$$

where ρ is the fluid density.

- **Momentum balance**

States the conservation of linear and angular momentum. For the momentum conservation equation, the external body forces \mathbf{f}_e acting on the system and the stress tensor $\boldsymbol{\sigma}$ act respectively as the volume and the surface source terms:

$$\frac{\partial \rho \mathbf{U}}{\partial t} + \nabla \cdot (\rho \mathbf{U} \mathbf{U}) = \nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{f}_e \quad (2.4)$$

The stress tensor can be split into two types of contributions, i.e., pressure and viscous forces:

$$\sigma_{ij} = (-p + 2\lambda \nabla \cdot \mathbf{U}) \delta_{ij} + \tau_{ij} \quad (2.5)$$

where λ is the second coefficient of viscosity (dilatational coefficient) and $\boldsymbol{\tau}$ is the viscous stress tensor, which depends on the fluid nature. In the present work only Newtonian fluids will be considered. In such case, $\boldsymbol{\tau}$ can be expressed as:

$$\tau_{ij} = 2\mu S_{ij} + \lambda S_{kk} \delta_{ij} \quad (2.6)$$

being μ the dynamic viscosity of the fluid and S_{ij} the rate of strain tensor, defined as:

$$S_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \quad (2.7)$$

The hydrostatic part of σ_{ij} can be then rearranged as:

$$\frac{1}{3}\sigma_{kk} = -p + \lambda S_{kk} + \frac{2}{3}\mu S_{kk} \quad (2.8)$$

In case of incompressible flows the velocity field \mathbf{U} is divergence-free ($\nabla \cdot \mathbf{U} = 0$) and Expression (2.8) simplifies as:

$$\frac{1}{3}\sigma_{kk} = -p \quad (2.9)$$

In case of compressible flows $\nabla \cdot \mathbf{U} \neq 0$ and, generally, the *Stokes hypothesis* is assumed:

$$\lambda + \frac{2}{3}\mu = 0 \quad (2.10)$$

The viscosity is therefore present only in the deviatoric component of $\boldsymbol{\sigma}$ and its hydrostatic component is equal to the thermodynamic pressure p . From all these considerations, the usual form of the momentum equation can be derived, which reads:

$$\frac{\partial \rho \mathbf{U}}{\partial t} + \nabla \cdot (\rho \mathbf{U} \mathbf{U}) = -\nabla p + \nabla \cdot (\mu \mathbf{S}) + \rho \mathbf{f}_e \quad (2.11)$$

• Energy balance

States the conservation of energy. The way the energy equation is formulated in the Navier-Stokes equations is not unique. Expressions exist for enthalpy, internal energy, temperature, pressure, total energy and entropy. Nevertheless, the only way to formulate this equation in conservative form is in terms of total energy. The conservative formulation is mandatory for capturing possible discontinuities of the flow and to properly correct the velocity in numerical simulations, as explained in [9].

Neglecting the contribution of the gravitational forces, the total energy E is the sum of internal energy e and kinetic energy K :

$$E = e + K, \quad \text{with } K = \frac{1}{2}(\mathbf{U} \cdot \mathbf{U}) \quad (2.12)$$

A transport equation for the internal energy only can be written as follows [10]:

$$\frac{\partial \rho e}{\partial t} + \nabla \cdot (\rho e \mathbf{U}) = -\nabla \cdot \mathbf{q} + \boldsymbol{\sigma} : \nabla \mathbf{U} + S_e \quad (2.13)$$

where \mathbf{q} is the heat flux, $\boldsymbol{\sigma} : \nabla \mathbf{U}$ represents a source of internal energy due to deformation work on the fluid particle done by viscous stresses and S_e is a generic internal energy source.

A transport equation for the kinetic energy K can be formulated in the following form:

$$\frac{\partial \rho K}{\partial t} + \nabla \cdot (\rho K \mathbf{U}) = \nabla \cdot (\boldsymbol{\sigma} \cdot \mathbf{U}) + S_K \quad (2.14)$$

where $\nabla \cdot (\boldsymbol{\sigma} \cdot \mathbf{U})$ represents the rate of change of strain energy and can be expressed as:

$$\nabla \cdot (\boldsymbol{\sigma} \cdot \mathbf{U}) = (\nabla \cdot \boldsymbol{\sigma}) \cdot \mathbf{U} + \boldsymbol{\sigma} : \nabla \mathbf{U} \quad (2.15)$$

Combining Equations (2.13), (2.14) and (2.15) it is possible to write the transport equation for the total energy E :

$$\frac{\partial \rho e}{\partial t} + \nabla \cdot (\rho e \mathbf{U}) + \frac{\partial \rho K}{\partial t} + \nabla \cdot (\rho K \mathbf{U}) = -\nabla \cdot (p \mathbf{U}) - \nabla \cdot \mathbf{q} + \nabla \cdot (\boldsymbol{\tau} \cdot \mathbf{U}) + S_e \quad (2.16)$$

If enthalpy is preferred over internal energy, Equation (2.16) can be reformulated, considering that $h = e + p/\rho$:

$$\frac{\partial \rho h}{\partial t} + \nabla \cdot (\rho h \mathbf{U}) + \frac{\partial \rho K}{\partial t} + \nabla \cdot (\rho K \mathbf{U}) = \frac{\partial p}{\partial t} - \nabla \cdot \mathbf{q} + \nabla \cdot (\boldsymbol{\tau} \cdot \mathbf{U}) + S_h \quad (2.17)$$

being S_h a generic enthalpy source.

• Constitutive relations

For compressible flows the relation between density, pressure and temperature is given by a special equation called *Equation of State*. The most common one is the ideal gas relation:

$$p = \rho R T \quad (2.18)$$

where T is the fluid temperature and R is the ideal gas constant.

For an ideal gas it is also possible to use the following thermodynamic relations to relate enthalpy and internal energy to temperature, so that the energy equation can be reformulated with temperature being the only unknown:

$$dh = c_p dT, \quad de = c_v dT \quad (2.19)$$

Furthermore, the variation of the dynamic viscosity with temperature will be accounted for by using the Sutherland's law:

$$\frac{\mu(T)}{\mu(T_0)} = \left(\frac{T}{T_0} \right)^{\frac{3}{2}} \frac{T_0 + S_1}{T + S_1}, \quad \text{with } S_1 = 110.4 \text{ K} \quad (2.20)$$

which is valid from 100 K to 1900 K [11]. To be noted that the use of this law introduces an additional non-linearity in the momentum and energy equations.

2.2.3 Governing equations for incompressible flows

In case of incompressible flows, the velocity field \mathbf{U} is solenoidal, i.e. divergence free. Furthermore, if the flow is isothermal, the energy balance equation is not needed anymore. In such cases, as the ones treated in Chapter 4, the Navier-Stokes equations set can be conveniently simplified as follows:

- **Mass balance**

$$\nabla \cdot \mathbf{U} = 0 \quad (2.21)$$

- **Momentum balance**

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot (\mathbf{U}\mathbf{U}) = -\frac{1}{\rho} \nabla p + \nabla \cdot (\nu \mathbf{S}) + \mathbf{f}_e \quad (2.22)$$

with ν being the kinematic viscosity of the fluid.

2.3 LES modeling

The basic LES modeling framework was introduced by Smagorinsky [12] in the early 1960's for the computation of geophysical flows. At that time computational resources were extremely limited so that a complete resolution of all the scales of motion was not a viable option. An alternative approach has been then inspired by Kolmogorov's theory [13].

Accordingly to this theory, for flows at sufficiently high Reynolds numbers, the small scale turbulent motions are statistically isotropic and universal and they act mainly draining energy from the larger scales through the energy cascade process. The idea at the basis of LES modeling has been therefore to directly compute the larger scales of motion, which contain most of the energy and are strongly affected by the boundary conditions, while the smaller scales could be conveniently represented by a model. In this way the computational cost is reduced by reducing the range of length and time scales that have to be resolved.

2.3.1 LES filtering operations

In order to separate the large scales of motion from the small ones a filtering operation is performed. The filter can be conceived as a local weighted average of the flow properties over a volume of fluid. If the turbulent flow is dominated by convective phenomena and is not influenced by any input of external energy at high frequency and low wavenumber (e.g. high frequency wall oscillations), the application of a space filter yields also the removal of the small flow time scales.

One of the key properties of the filtering process is the filter width, Δ : motion at scales larger than Δ (resolved or Grid-Scale (GS)) is retained in the filtered flow field, while the contributions of scales smaller than Δ (Sub-Grid Scale (SGS)) are effectively removed from the solution. Such contributions are not irrelevant and their effect on the resolved flow has to be modeled through a proper SGS model. The function of the SGS model is crucial for problems in which small-scale contributions play a significant role, such as boundary layer flows, reacting flows and multiphase flows.

In LES any flow variable $f(\mathbf{x}, t)$ can be formally decomposed in a large scale (or resolved, or GS) term, $\bar{f}(\mathbf{x}, t)$, and in a small scale (or SGS) term, $f'(\mathbf{x}, t)$:

$$f(\mathbf{x}, t) = \bar{f}(\mathbf{x}, t) + f'(\mathbf{x}, t) \quad (2.23)$$

The filtering operation is performed through a mathematical operation

of convolution [6]:

$$\bar{f}(\mathbf{x}, t) = \int_D G_\Delta(\mathbf{x}, \mathbf{x}') f(\mathbf{x}, t) d\mathbf{x}' \quad (2.24)$$

where D is the computational domain and $G_\Delta(\mathbf{x}, \mathbf{x}')$ is the filter kernel. $G_\Delta(\mathbf{x}, \mathbf{x}')$ is a localized function, i.e. a function that is non-zero only when \mathbf{x}' lies in the neighborhood of \mathbf{x} , and satisfies the following normalization condition:

$$\int_D G_\Delta(\mathbf{x}, \mathbf{x}') d\mathbf{x}' = 1 \quad (2.25)$$

Figure 2.2 shows a graphical representation of the filter process where a generic filter kernel is applied to a randomly fluctuating variable f .

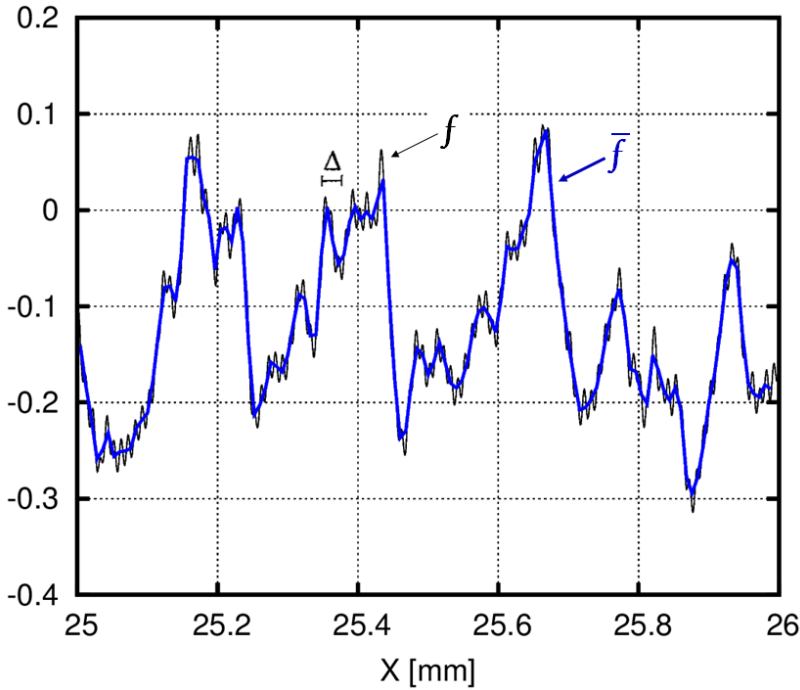


Figure 2.2: Filtering of a randomly fluctuating variable f .

Several filter functions have been proposed during the years (see [14] for reference). Amongst these, the most popular ones are the following:

- Gaussian filter

It has the desirable properties of smoothness and differentiability:

$$G_{\Delta}(\mathbf{x}, \mathbf{x}') = \left(\frac{6}{\pi \Delta^2} \right)^{\frac{1}{2}} \exp\left(- \frac{6|\mathbf{x} - \mathbf{x}'|^2}{\Delta^2} \right) \quad (2.26)$$

- Sharp spectral cutoff filter

It is defined as:

$$G_{\Delta}(\mathbf{x}, \mathbf{x}') = \prod_{i=1}^3 \frac{\sin\left(\frac{\pi}{\Delta}(x - x')\right)}{\frac{\pi}{\Delta}(x - x')} \quad (2.27)$$

or, in spectral space, as:

$$\hat{G}_{\Delta}(k) = \begin{cases} 1 & \text{if } k \leq \frac{\pi}{\Delta}, \\ 0 & \text{otherwise.} \end{cases} \quad (2.28)$$

This filter eliminates all Fourier modes of wavenumber greater than the cutoff wavenumber (π/Δ) while it has no effects on the lower wavenumbers of the solution. Being non-local in physical space, it is difficult to apply to inhomogeneous flows and it is normally used in conjunction with spectral methods.

- Box filter (or top-hat filter)

It is defined as:

$$G_{\Delta}(\mathbf{x}, \mathbf{x}') = \begin{cases} \frac{1}{\Delta^3} & \text{if } |\mathbf{x} - \mathbf{x}'| \leq \frac{\Delta}{2}, \\ 0 & \text{otherwise.} \end{cases} \quad (2.29)$$

It is basically an average over a cubic region of volume Δ^3 . The box filter is the most widely adopted filter in Finite Volume implementations of LES. For these implementations, the formulation of the filter implies that, if the filter width Δ is chosen to be equal to the grid-spacing, the local value of f will coincide with its filtered value \bar{f} .

Despite the abundance of filter formulations, it has to be noted that in many practical applications the so-called implicit filtering approach is adopted. This is certainly true for many commercial codes (*AVL Fire*[®], *ANSYS Fluent*[®], *ANSYS CFX*[®], *ADAPCO StarCCM+*[®], just to cite some) and for *OpenFOAM*[®] as well. Indeed, in the implicitly filtered LES, the

discretization of both domain and differential operators acts, in practice, as a built-in box filter, without any need to explicitly perform any filtering operation. More details on this last non-trivial aspect of practical LES applications can be found in [15].

2.3.2 The filtered Navier-Stokes equations

In order to retrieve the filtered Navier-Stokes equations set for LES computations, the governing equations of Section 2.2 have been considered. For sake of simplicity, here the incompressible formulation of Paragraph 2.2.3 has been used and no volume forces have been considered. Using an overbar to denote the filtering operation and adopting the Einstein notation, the filtered NS equations read:

$$\frac{\partial \bar{u}_i}{\partial t} + \frac{\partial(\bar{u}_i \bar{u}_j)}{\partial x_j} = -\frac{1}{\rho} \frac{\partial \bar{p}}{\partial x_i} + \nu \frac{\partial^2 \bar{u}_i}{\partial x_j \partial x_j} - \frac{\partial \tau_{ij}^{SGS}}{\partial x_j} \quad (2.30)$$

and the fluid incompressibility constraint is expressed by:

$$\frac{\partial \bar{u}_i}{\partial x_i} = 0 \quad (2.31)$$

where u , p , ρ and ν are respectively the fluid velocity, pressure, density and (molecular) kinematic viscosity.

To be noted that, if the compressible formulation has to be used, the Favre filtering is performed, i.e. a change of variable is made so that the filtered variables are weighted by the density.

The filtered equations set (2.30)-(2.31) is formally similar to the original NS equations set, except for the term containing the sub-grid stresses defined as:

$$\tau_{ij}^{SGS} = \overline{u_i u_j} - \bar{u}_i \bar{u}_j \quad (2.32)$$

This term vanishes if the grid density is sufficiently high. In such case the filter width is small enough and all scales of motion are resolved so that the LES approaches a DNS.

The sub-grid stress tensor τ_{ij}^{SGS} is functionally similar to the Reynolds stress in RANS modeling, but represents a significantly smaller part of the turbulent energy spectrum than the RANS turbulent energy, so that the accuracy of the stress model is somehow less crucial than in RANS computation. Nevertheless, if the turbulent dissipation in the small scales is not accurately predicted, it can lead to energy accumulation in the resolved scales and, potentially, to computational instabilities.

In order to better understand the nature of the SGS stress tensor, this can be decomposed by using Equation (2.23). From this operation three distinct terms arise:

$$\tau_{ij}^{SGS} = \overline{u_i u_j} - \overline{u_i} \overline{u_j} = (\overline{\overline{u_i u_j}} - \overline{u_i} \overline{u_j}) + (\overline{\overline{u_i u_j'}} + \overline{u_i' \overline{u_j}}) + \overline{u_i' u_j'} \quad (2.33)$$

On the RHS of the expression, the three contributions have distinct physical meanings:

- Leonard stresses L_{ij}

$$L_{ij} = \overline{\overline{u_i u_j}} - \overline{u_i} \overline{u_j} \quad (2.34)$$

The Leonard stresses L_{ij} can be directly computed from the resolved velocity field [16]. They are caused by the application of a second filtering operation that changes the originally filtered flow variable. It has in fact to be noted that unlikely in time-averaging, for space-filtering $\overline{\overline{u}} \neq \overline{u}$. These stresses represent the contribution of the interactions between resolved eddies in the production of sub-grid turbulence.

- Cross stresses C_{ij}

$$C_{ij} = \overline{\overline{u_i u_j'}} + \overline{u_i' \overline{u_j}} \quad (2.35)$$

The cross stresses C_{ij} represent the interactions between the small scale eddies and the resolved ones. An approximate formulation for these contributions have been proposed by Ferziger in [17].

- SGS Reynolds stresses R_{ij}

$$R_{ij} = \overline{u_i' u_j'} \quad (2.36)$$

The SGS Reynolds stresses R_{ij} are due to interactions between the SGS eddies. No suitable approximations are available for these stresses.

Unfortunately the correlations that have been proposed to model these stresses contain several approximations that do not allow precise evaluations. Furthermore, while the τ_{ij}^{SGS} as a whole is Galilean invariant (i.e., it is independent of the chosen inertial frame), the Cross and Leonard stresses are not. Due to these reasons, the approaches following the decomposition of Equation (2.33) have nowadays been abandoned, preferring to model τ_{ij}^{SGS} as a whole by using a SGS model.

2.3.3 Sub-Grid Scale modeling

Despite the lack of a universally accepted SGS turbulence model for LES, the eddy-viscosity assumption is, to date, the most common modeling approach, at least for industrial applications. The SGS stress tensor τ_{ij}^{SGS} can be then expressed as:

$$\tau_{ij}^{SGS} - \frac{1}{3}\tau_{kk}^{SGS}\delta_{ij} = -2\nu_{SGS}\overline{S_{ij}} \quad (2.37)$$

where $\overline{S_{ij}}$ is the resolved rate-of-strain tensor defined as:

$$\overline{S_{ij}} = \frac{1}{2}\left(\frac{\partial \overline{u}_i}{\partial x_j} + \frac{\partial \overline{u}_j}{\partial x_i}\right) \quad (2.38)$$

With the eddy-viscosity assumption, the formulation commonly adopted for ν_{SGS} is the following:

$$\nu_{SGS} = (C_m\Delta)^2\overline{OP}(\vec{x}, t) \quad (2.39)$$

where C_m is the model constant, Δ is the filter width and \overline{OP} is an operator specific of each sub-grid model.

One of the earliest and most widely adopted SGS model is the Smagorinsky model [12]. This algebraic model has been derived assuming that the small eddies are in equilibrium and dissipate completely and instantaneously the energy received from the interaction with the resolved eddies. For this model the operator \overline{OP} reads:

$$\overline{OP} = \sqrt{2\overline{S_{ij}}\overline{S_{ij}}} \quad (2.40)$$

The model constant has been determined by evaluating the decay rate of isotropic turbulence [18], which suggested values in the range of $C_S = 0.17 \div 0.21$. Nevertheless, other studies performed on channel flows suggested that smaller values of the constant C_S have to be used for internal flows computations ($C_S \approx 0.1$). The difference in the constant values has been attributed to the effect of the mean flow strain and shear and has suggested that more sophisticated SGS approaches have to be used to pursuit successful LES modeling.

Several types of SGS models have been developed through the years. They can be generally categorized as:

- Algebraic models
- Differential models

- Scale similarity and mixed models
- Dynamic models

It is not in the scope of the current work to discuss all the different model types, for which a wide literature is available (see, for instance [14], [19] or [20]). Within this research, only algebraic and dynamic models have been employed, choosing the ones best suited for ICE flows and favoring characteristics such as robustness, simplicity and (computational) cheapness.

Furthermore, this research will not consider the Implicit LES (or ILES) approach, in which no explicit SGS model is applied. ILES relies instead on the intrinsic dissipation of the numerical scheme to act as a SGS model. The interested reader is referred to [21] or [22] for details regarding this technique.

2.4 Implemented sub-grid scale models

In the following paragraphs the basic details of the tested SGS models will be recalled. The dynamic Smagorinsky model was already present in the *OpenFOAM*[®] libraries but it has been modified in order to compute additional quantities of interest for the current investigation, such as k_{SGS} or ε_{SGS} . All the other SGS models here presented were not available in the libraries and they have been wholly implemented during the research activity.

2.4.1 The dynamic Smagorinsky SGS model

The dynamic Smagorinsky model is based on the dynamic procedure developed by Germano [23] and subsequent improvement proposed by Lilly [24]. The model employs a second filter, the test filter $\hat{\Delta}$ larger than the initial filter Δ and usually chosen as $\hat{\Delta} = 2\Delta$. The tensor operator \overline{OP} in Equation (2.35) is based on the second invariant of the strain rate tensor $\overline{S_{ij}}$ of the filtered velocity field \overline{u}_i :

$$\overline{OP} = \sqrt{2\overline{S_{ij}}\overline{S_{ij}}} \quad (2.41)$$

The model constant is then dynamically evaluated from test-filtered resolved quantities:

$$C_{dS}^2 = \frac{1}{2} \frac{\langle L_{ij}M_{ij} \rangle^+}{\langle M_{ij}M_{ij} \rangle} \quad (2.42)$$

where L_{ij} is the Leonard stress tensor:

$$L_{ij} = \widehat{\overline{u}_i \overline{u}_j} - \widehat{\overline{u}_i} \widehat{\overline{u}_j} \quad (2.43)$$

and

$$M_{ij} = \hat{\Delta}^2 \widehat{\overline{OP} \overline{S_{ij}}} - (\Delta^2 \widehat{\overline{OP} \overline{S_{ij}}}) \quad (2.44)$$

The superscript + in Equation (2.42) means a positive clipping of all negative values to zero and the brackets $\langle \rangle$ denote a volume averaging operation adopted in order to avoid numerical instability. To be noted that in the present work homogeneous volume averaging is employed.

2.4.2 The WALE SGS model

The WALE model has been developed by Nicoud and Ducros [25] and its invariant is based on the square of the velocity gradient tensor. In particular, the operator \overline{OP} is based on the traceless symmetric part of the square of the velocity gradient tensor S_{ij}^d , defined as

$$\overline{S_{ij}^d} = \frac{1}{2}(\overline{g_{ij}^2} - \overline{g_{ji}^2}) - \frac{1}{3}\delta_{ij}\overline{g_{kk}^2} \quad (2.45)$$

where $\overline{g_{ij}}$ is the velocity gradient tensor. The tensor $\overline{S_{ij}^d}$ can also be expressed as:

$$\overline{S_{ij}^d} = \overline{S_{ik}}\overline{S_{kj}} + \overline{\Omega_{ik}}\overline{\Omega_{kj}} - \frac{1}{3}\delta_{ij}[\overline{S_{mn}}\overline{S_{mn}} - \overline{\Omega_{mn}}\overline{\Omega_{mn}}] \quad (2.46)$$

being $\overline{S_{ij}}$ and $\overline{\Omega_{ij}}$ respectively the rate of strain and the vorticity tensors of the resolved velocity field. By construction the trace of $\overline{S_{ij}^d}$ is zero and its second invariant remains finite and proportional to $\overline{S_{ij}^d}\overline{S_{ij}^d}$. Assuming incompressibility, this latter term can be expressed as:

$$\overline{S_{ij}^d}\overline{S_{ij}^d} = \frac{1}{6}(\overline{S^2S^2} + \overline{\Omega^2\Omega^2}) + \frac{2}{3}\overline{S^2\Omega^2} + 2IV_{S\Omega} \quad (2.47)$$

where $\overline{S^2} = \overline{S_{ij}S_{ij}}$, $\overline{\Omega^2} = \overline{\Omega_{ij}\Omega_{ij}}$ and $IV_{S\Omega} = \overline{S_{ik}}\overline{S_{kj}}\overline{\Omega_{jl}}\overline{\Omega_{li}}$.

Thanks to the operator $\overline{S_{ij}^d}\overline{S_{ij}^d}$ the WALE model is sensitive to both strain and rotation rates. The aforementioned operator is zero in case of pure shear and this means that unlike the Smagorinsky model, the WALE model avoids producing eddy-viscosity in the case of wall-bounded laminar flows, resulting in an accurate reproduction of the laminar to turbulent transition. The tensor operator \overline{OP} , Equation (2.35), in the WALE model reads then:

$$\overline{OP}^W = \frac{(\overline{S_{ij}^d}\overline{S_{ij}^d})^{3/2}}{(\overline{S_{ij}S_{ij}})^{5/2} + (\overline{S_{ij}^d}\overline{S_{ij}^d})^{5/4}} \quad (2.48)$$

The model constant has been set to $C_W = 0.575$ as an average of the values reported in [25]. When the distance from a solid wall approaches zero, the invariant (2.47) scales as y^2 so that the WALE operator \overline{OP}^W scales as y^3 reproducing the proper behavior of the eddy-viscosity in near-wall regions.

2.4.3 The dynamic WALE SGS model

The dynamic procedure proposed by Germano [23] to develop the dynamic Smagorinsky SGS model can be applied also to the WALE model. In such case the Expression (2.42) is replaced by:

$$C_{dW}^2 = \frac{1 < L_{ij} M_{ij}^W >^+}{2 < M_{ij}^W M_{ij}^W >} \quad (2.49)$$

where L_{ij} is the same as (2.43) and the superscript + denotes the usual positive-clipping procedure. The operator M_{ij}^W then reads:

$$M_{ij}^W = \widehat{\Delta^2 \overline{OP^W S_{ij}}} - (\Delta^2 \overline{OP^W S_{ij}}) \quad (2.50)$$

As pointed out by Baya Toda et al. [26] the dynamic version of the WALE model produces excessively high values of the constant C_{dW} at the walls and a poor prediction of the mean velocity. The reason for this has been found by the Authors [26] who demonstrated that the dynamic procedure tends to alter the proper near-wall y^3 behavior of the model constant leading to a y^0 scaling in this region. Since the laminar viscosity is predominant in the proximity of solid boundaries, they concluded that there is no reason to dynamically adapt the model constant there. In order to identify the near-wall region it has been developed a sensor capable of detecting the presence of a wall with no need to have an a-priori knowledge of the geometry, that can be arbitrary complex. The sensor has been named "Shear and Vortex Sensor" (SVS) and has the following formulation:

$$SVS = \frac{(\overline{S_{ij}^d S_{ij}^d})^{3/2}}{(\overline{S_{ij}^d S_{ij}^d})^{3/2} + (\overline{S_{ij} S_{ij}})^3} \quad (2.51)$$

The SVS invariant is bounded between 0 and 1, where the lower value corresponds to pure shear and the higher corresponds to pure rotating flows. Thanks to this property the computational domain can be ideally split into two regions: the near-wall region, identified by $SVS < 0.09$ (as proposed by the Authors in [26]), where the model constant is fixed at its standard value, and a bulk region where the model constant is evaluated dynamically. The (modified) dynamic WALE SGS model can be summarized as follows:

- if $SVS < 0.09$:

$$C_{dW}^2 = \frac{1 < L_{ij} M_{ij}^W >^+}{2 < M_{ij}^W M_{ij}^W >} \quad (2.52)$$

- if $SVS \geq 0.09$:

$$C_{dW}^2 = 0.25 \quad (2.53)$$

2.4.4 The Sigma SGS model

The Sigma SGS model has been recently proposed by Baya Toda et al. [27] as an upgrade of the WALE model.

In [27], the Authors highlight some deficiencies of the WALE model: it can be shown that it produces non-zero eddy viscosity in simple flow configuration such as solid rotation and axisymmetric expansion or contraction, where the sub-grid scale activity should ideally be zero. The model has been developed in order to force the operator \overline{OP} to have the following properties:

- P1: to behave as y^3 in proximity of walls.
- P2: to vanish in the case of solid body flow rotation (like the Smagorinsky model) and in the case of pure shear (like the WALE and Vreman [28] models). In general, the operator should vanish in the case of any two-dimensional flow.
- P3: to vanish where the resolved flow field is either in a pure axisymmetric or in a isotropic contraction/expansion configuration.

In the Sigma model the operator \overline{OP} takes the following form:

$$\overline{OP}^\sigma = \frac{\sigma_3(\sigma_1 - \sigma_2)(\sigma_2 - \sigma_3)}{\sigma_1^2} \quad (2.54)$$

where σ_1, σ_2 and σ_3 are the three singular values of the velocity gradient tensor \bar{g}_{ij} . By definition these values are always positive and equal to the square root of the eigenvalues of the matrix $\mathbf{G} = \mathbf{g}^t \mathbf{g}$, which themselves are always positive because \mathbf{G} is symmetric semi-definite positive. Analyzing the asymptotic behavior of the singular values at walls, it can be shown that:

$$\sigma_1 = O(y^0); \sigma_2 = O(y^1); \sigma_3 = O(y^2) \quad (2.55)$$

Thanks to the relations (2.55) it is possible to verify that in Equation (2.54), the product $\sigma_3(\sigma_1 - \sigma_2)(\sigma_2 - \sigma_3)$ satisfies the above mentioned properties P1, P2 and P3. The term σ_1^2 at denominator has been chosen in order to scale \overline{OP}^σ to a frequency scale. The model constant is set $C_\sigma = 1.5$ as proposed in [27].

The source code of the implemented models is reported in Appendix A

2.5 The Finite Volume Method

2.5.1 Numerical methods for CFD simulations

Several numerical methods exist for the solution of CFD problems. These include the well known Finite Volume (FV), Finite Element (FE), Finite Difference (FD) and Spectral Methods (SM) along with more exotic approaches such as Particle Methods (PM) and Lattice-Boltzmann Automata (LBA). All these methods have been, more or less, successfully employed for LES simulations.

Academic research largely prefer to employ high-order methods (third order or above) so that FD and SM are the most common approaches for such applications since they allow to easily and reliably reach high accuracy levels. However, they require body-fitted structured grids in the case of FD or computational domains with homogeneous directional discretizations in the case of SM, thus severely limiting their application to academic test-cases.

Conversely, the FE method has no such constraints in domain discretization, while retaining high-order accuracy capabilities. Despite this, the method's formulation does not guarantee the conservation of conservative quantities such as mass, momentum or energy, thus jeopardizing the fidelity of CFD simulations. In fact, conservativeness is one of the essential features that a CFD code must present and it is certainly much more important than the formal order of accuracy of the chosen method of solution.

The FV method, which is nowadays the most widely adopted method in commercial CFD codes, represents the best compromise between simulation accuracy and the aforementioned domain discretization issues. Conservativeness of the method is intrinsic in its formulation and it allows handling arbitrarily complex grids. The FV method is arguably the simplest method to understand and to code, and this explain why it is so popular among engineers.

2.5.2 The Finite Volume discretization process

The FV discretization process is performed in two fundamental steps:

- In the first step the physical domain is discretized into a finite number of contiguous, non-overlapping computational convex cells. Such cells together form a computational grid which is a discrete representation of the computational domain. If the physical domain moves or changes its topology in time, so will do its computational counterpart, with cells moving and/or modifying their connectivity.

- In the second step the equations governing the fluid motion are approximated onto the computational cells (space discretization) for each time-step (time discretization). Such equations, presented in Section 2.2 can be rearranged in the following general form:

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\mathbf{U}\phi) = +\nabla \cdot (\Gamma \nabla \phi) + \mathbf{S} \quad (2.56)$$

where ϕ is the generic variable to be solved such as a momentum component, the density, the energy or a turbulent quantity and Γ is the diffusivity associated to such variable (e.g. a viscosity, in the case of momentum). To be noted that accordingly to the pressure-velocity coupling chosen, the continuity equation could be handled differently, as explained in [8] or [10].

Equation (2.56) is then integrated over a generic CV and rearranged using the Gauss theorem:

$$\int_V \frac{\partial \phi}{\partial t} dV + \int_{\Omega} \mathbf{U}\phi \cdot \mathbf{n} dA = \int_{\Omega} \Gamma \nabla \phi \cdot \mathbf{n} dA + \int_V \mathbf{S} dV \quad (2.57)$$

where V and Ω are respectively the volume and the (surface) boundary of the CV, \mathbf{n} is the face normal unit vector and A is the cell face area. The CV topology is reported in Figure 2.3.

In a colocated arrangement [10], such as the one used by *OpenFOAM*[®], the variable ϕ is stored at each mesh cell centroid and the variable value for the cell P will be denoted with ϕ_P . The cell P is surrounded by its neighboring cells: N, E, S, W, which stands respectively for *North*, *East*, *South* and *West*. Each of these cells has its own value of the variable ϕ , respectively ϕ_N , ϕ_E , ϕ_S and ϕ_W . The computational molecule is completed by an outermost cell to the left side, WW, with its corresponding variable value ϕ_{WW} . The cell P shares faces with its surrounding cells. For instance, face n is the face shared between P and N, face e is the face shared between P and E, and so on.

Equation (2.57) has four terms which need to be discretized:

- $\int_V \frac{\partial \phi}{\partial t} dV \Rightarrow$ the temporal variation term
- $\int_{\Omega} \mathbf{U}\phi \cdot \mathbf{n} dA \Rightarrow$ the convective term
- $\int_{\Omega} \Gamma \nabla \phi \cdot \mathbf{n} dA \Rightarrow$ the diffusive term
- $\int_V \mathbf{S} dV \Rightarrow$ the source term

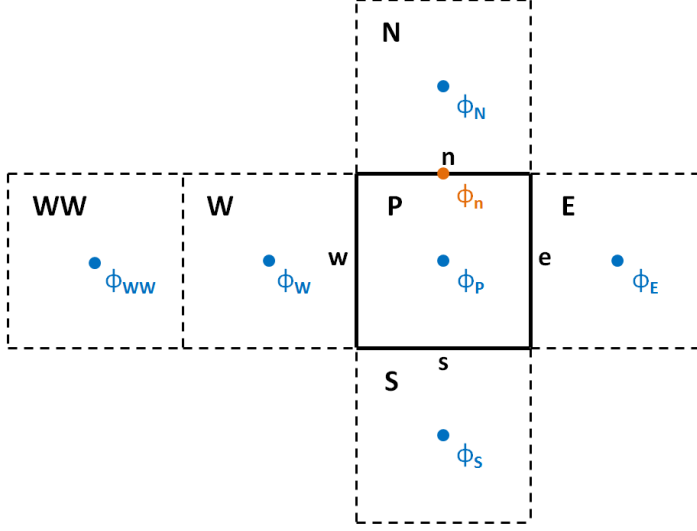


Figure 2.3: Control volume topology.

Each of these terms have different properties and need different discretization techniques. More details will be given in the sections that follow.

2.5.3 Time discretization

The time derivative present in Equation (2.57) can be treated in many different ways. Here are the three most common discretization approaches:

- **First-order backward difference scheme**

The time derivative present in Equation (2.57) can be written as:

$$\frac{\partial \phi}{\partial t} \approx \frac{\phi^n - \phi^{n-1}}{\Delta t} \quad (2.58)$$

where ϕ^n and ϕ^{n-1} are respectively the variable ϕ values at the n^{th} and at $(n - 1)^{th}$ timestep, while Δt is the timestep duration (usually assumed constant). This approach, also known as *Euler backward scheme*, is implicit, unconditionally stable and guarantees a first-order accuracy in time.

- **Second-order backward difference scheme**

For better accuracy, the time derivative can be approximated as:

$$\frac{\partial \phi}{\partial t} \approx \frac{3\phi^n - 4\phi^{n-1} + \phi^{n-2}}{2\Delta t} \quad (2.59)$$

obtaining an implicit second-order accurate scheme. Despite its greater accuracy, it is conditionally stable and can produce oscillatory solutions if large values of the timestep are employed.

- **Crank-Nicolson blended scheme**

A more robust solution could be obtained with the *Crank-Nicolson scheme* [29]. This scheme represents a compromise between the first and the second order backward difference scheme: it is unconditionally stable and less prone to oscillations but the accuracy lies in between the two, depending on the value of a blending parameter α : for $\alpha = 0$ the Euler scheme is recovered, while for $\alpha = 1$ the second-order backward scheme is obtained.

2.5.4 Space discretization

The following paragraphs describe the space discretization methods used to approximate the integral terms found in Equation (2.57).

- **Face interpolation**

The interpolation of the cell-centered values of the variable ϕ onto the cell face centers has paramount importance for the whole discretization procedure since it is used in the approximation of several terms of Equation (2.57). To date, several interpolation schemes have been formulated and a review of the main ones is available in [30], [31] or [32].

In order to compute the face value ϕ_f of the variable ϕ , the values of the neighboring cells are needed. Moreover, in some cases the computation requires the evaluation of a flux F through the face f . The definition of the flux F depends on the nature of the equation treated. For instance, the mass flux through face f of (vector) surface \mathbf{A} , being $\mathbf{A} = nA$, is expressed as $F = \mathbf{A} \cdot (\rho \mathbf{U})_f$.

Common choices for the face interpolation schemes are the following:

- **Upwind differencing - UD**

This scheme computes ϕ_f depending on the flow direction [10]:

$$\phi_{f,UD} = \begin{cases} \phi_P & \text{for } F \geq 0 \\ \phi_N & \text{for } F < 0 \end{cases} \quad (2.60)$$

Despite guaranteeing the boundedness of the solution, the UD provides first-order accuracy only. In [8] it is shown how this scheme is affected by a leading truncation error resembling a diffusive flux. Such numerical diffusion can severely affect solution accuracy on coarse grids so that UD has to be avoided for LES simulations. It can be however a valuable choice for flow initialization purposes.

– Central differencing - CD

The CD scheme computes ϕ_f through a linear combination of the cell-center values ϕ_P and ϕ_N :

$$\phi_{f,CD} = g_f \phi_P + (1 - g_f) \phi_N \quad (2.61)$$

with the weight g_f being:

$$g_f = \frac{|\mathbf{d}_{fN}|}{|\mathbf{d}_{fN}| + |\mathbf{d}_{fP}|} \quad (2.62)$$

where \mathbf{d}_{fN} is the distance vector between face f center and cell N center, whereas \mathbf{d}_{fP} is the distance vector between face f center and cell P center. This scheme provides second-order accuracy but does not guarantee solution boundedness [10]. It can be reliably employed only on very regular and refined grids and/or low-Reynolds diffusion dominated flows. It is therefore not well suited for industrial applications.

– Blended differencing - BD

In order to overcome the limitations of the aforementioned schemes, the BD scheme has been devised, attempting to guarantee boundedness while preserving a reasonable accuracy. The BD scheme is basically a linear combination of UD and CD:

$$\phi_{f,BD} = (1 - \gamma) \phi_{f,UD} + \gamma \phi_{f,CD} \quad (2.63)$$

where γ is the blending factor that determines the amount of numerical diffusion introduced in the solution. When $\gamma = 0$ a pure UD scheme is obtained, while for $\gamma = 1$ a pure CD scheme is retrieved.

Several formulations have been proposed for the blending factor γ attempting to provide boundedness while reducing as much as possible numerical diffusion, which is particularly important for the discretization of the convection term in transport equations.

The most promising of these attempts adopt local adjustment formulations, such as the *Total Variation Diminishing* (TVD) and the *Normalized Variable Diagram* (NVD) classes of schemes. These methods use an "unboundedness indicator" of some kind in order to detect the regions of the domain where intervention in the discretisation is needed. A thorough review of these schemes can be found in [10], [31] or [33].

Solution boundedness has great importance in CFD computations: any fluid property (such as temperature, density, phase fraction, etc.) must present values lying within proper bounds in order to be physically realistic and this should be guaranteed by the numerical scheme adopted.

All the simulations presented in this thesis have been carried out using a high order scheme: pure CD has been adopted for the test cases of Chapter 4, while the *Gamma* NVD scheme [33] has been employed for the flow bench LES simulations of Chapter 5.

• Gradient

The gradient operators in Equation (2.57) can be evaluated in basically two ways: applying the Gauss theorem, or through a least-square procedure. Despite both methods being available in *OpenFOAM*[®], only the former has been used in this study. The cell-centered gradient of ϕ is then obtained by applying Gauss integration:

$$\int_V \nabla \phi dV = \int_{\Omega} \phi \mathbf{n} dA \approx \sum_f \mathbf{n}_f A_f \phi_f \quad (2.64)$$

where ϕ_f can be computed through one of the methods described in the previous section.

If the face-normal gradient $\nabla_f^\perp \phi$ is needed, i.e. the product of the face gradient $\nabla_f \phi$ with the unit normal \mathbf{n} to the face, this can be evaluated using the following expression:

$$\nabla_f^\perp \phi = \frac{\phi_N - \phi_P}{|\mathbf{d}|} \quad (2.65)$$

where \mathbf{d} is the distance vector between the centers of cells P and N. To be noted that this approximation is second-order accurate only if \mathbf{d} is orthogonal to the face. In case of non-orthogonal grids, a correction term could be used to alleviate this accuracy deficiency.

- **Convective term**

The convective term in Equation (2.57) is discretized by applying the Gauss theorem, as usual:

$$\int_V \nabla \cdot (\phi U) dV = \int_{\Omega} (\phi U) \cdot \mathbf{n} dA \approx \sum_f \mathbf{n} A \cdot U_f \phi_f = \sum_f F U_f \phi_f \quad (2.66)$$

where F is an appropriate variable flux through the face f . In case of incompressible flows, such as the one Equation (2.57) refers to, F is the volumetric flux defined as $F = \mathbf{n} A \cdot \mathbf{U}_f$. Here again, the face value ϕ_f can be computed with any of the methods previously described.

- **Diffusive term**

The diffusive term in Equation (2.57) is again discretized through the Gauss theorem:

$$\int_V \nabla \cdot (\Gamma \nabla \phi) dV = \int_{\Omega} (\Gamma \nabla \phi) \cdot \mathbf{n} dA \approx \sum_f \Gamma_f (\mathbf{n} A \cdot \nabla_f \phi) \quad (2.67)$$

On orthogonal grids, the face normal gradient as defined in Equation (2.65) is a second order accurate approximation for the face gradient used here.

On non-orthogonal meshes, an additional correction term is needed in order to preserve second order accuracy. The gradient approximation is then computed using the following expression:

$$\mathbf{n} A \cdot \nabla_f \phi = |\Delta| \nabla_f^\perp \phi + \mathbf{k} \cdot (\nabla \phi_f) \quad (2.68)$$

where, on the RHS, the first term represents the orthogonal contribution and the second term represents the non-orthogonal correction. The vectors Δ and \mathbf{k} have to be determined accordingly to the non-orthogonality treatment adopted. To date, several approaches are available to treat non-orthogonality [8] and the one adopted in this work is based on the over-relaxation method proposed by Jasak in [32].

- **Source term**

The terms of Equation (2.57) that can't be written as convective, diffusive or temporal terms are usually treated as sources. The source

term, S , can be a general function of ϕ and before discretization it is linearized as:

$$S(\phi) = \phi S_I + S_E \quad (2.69)$$

where both S_I and S_E can be functions of ϕ . The source term so linearized is then integrated over the CV as:

$$\int_V S(\phi) dV = S_I V_P \phi_P + S_E V_P \quad (2.70)$$

Linearization is particularly important in implicit calculations, and care must be taken when performing it since its interaction with other equation's terms could adversely influence the solver convergence.

2.5.5 Solution of linear equations systems

The governing equations, properly discretized as described in the previous sections, are assembled for all CVs forming a system of algebraic equations. Such system will be linear or non-linear depending upon the nature of the partial differential equations from which it has been derived and will have the following form:

$$\mathbf{A}\mathbf{x} = \mathbf{b} \quad (2.71)$$

where \mathbf{A} is the coefficients matrix, \mathbf{x} is the system unknowns vector and \mathbf{b} is the system sources vector. These systems are usually very large, being their size proportional to the number of cells, N , of the computational domain. Every row of the system represents the mathematical connection of a cell with the others. Since usually each cell is directly connected to cells belonging to its neighborhood, only a few elements in the row will be non-zero, i.e., the system matrix \mathbf{A} will be *sparse*.

Once the assembly is performed, the system has to be solved and this can be done using either:

- **Direct Methods**

Direct methods solve the System (2.71) in a direct manner by rearranging the system matrix. Among several techniques, two of the most basic examples are the *Gaussian elimination* and the *LU decomposition*. Unfortunately, the number of operations required by these methods, and consequently the CPU time needed, scales with N^3 . Since for a typical CFD grid the number of cells N is very large (millions or

tens of millions for LES), direct approaches are computationally very expensive and rarely employed. Furthermore, the matrix system rearrangement tends to reduce the matrix sparsity with significant increase in memory allocation requirements [10].

- **Iterative Methods**

Iterative methods solve the System (2.71) by repeated application of simple algorithms leading to an eventual converge after a usually large number of iterations. The total number of operation required by an iterative method cannot be predicted in advance as for a direct method, but, if the convergence rate is decent, the former is often faster than the latter. Moreover, this class of methods require just to store the non-zero matrix elements, thus significantly alleviating memory needs.

In [8] Ferziger reformulates Equation (2.71) for an iterative solution in the following way:

$$\mathbf{A}\mathbf{x}^n = \mathbf{b} - \boldsymbol{\rho}^n \quad (2.72)$$

where \mathbf{x}^n is an approximate solution obtained after n iteration. Since \mathbf{x}^n doesn't satisfy exactly $\mathbf{A}\mathbf{x} = \mathbf{b}$, the residual vector $\boldsymbol{\rho}^n$ is introduced. The purpose of an iterative method is then to reduce such residual as much as possible, and, ideally, to bring it to machine zero.

To date, several iterative methods have been devised for CFD applications which differ in terms of complexity and efficiency. *Jacobi* and *Gauss-Seidel* are general-purpose basic iterative algorithms characterized by a slow convergence rate. Beyond these, more sophisticated and efficient options are the *Conjugate Gradient Methods* and the *Multigrid Methods*. *OpenFOAM*[®] provides numerical solvers of both these latter types allowing the user to select the most appropriate one depending on application.

A detailed description of the aforementioned methods would be beyond the scope of the present work and the interested reader is referred to [8], [10], [30] or [34] for further details.

2.5.6 Pressure-velocity coupling

The fluid flow Equations (2.3) - (2.16) constitute a set of conservation equations for mass, momentum and energy. Unfortunately, no analogous equation can be written for the pressure, p , that appears in the momentum equation.

In compressible flows, the continuity Equation (2.3) is used to compute the density and the pressure field can then be explicitly evaluated through an equation of state, such as Equation (2.18).

In incompressible flows, this approach is not feasible since the density is assumed constant and not linked to the pressure. In this case, the continuity equation represents a sort of kinematic constraint on the velocity field rather than a balance equation.

Due to the strong connection existing between pressure gradients and continuity, it is mandatory to construct a pressure field capable of guaranteeing continuity and this task is performed by a pressure-velocity coupling algorithm.

Pressure-velocity algorithms have been subject of intense research and more details on the many options available can be found in [8] or [10]. In *OpenFOAM*[®] four different methods are available to the user in the standard libraries: SIMPLE [35] and SIMPLEC [36] for steady-state flows plus PISO [37] and a hybrid SIMPLE/PISO [38] for transients.

Despite the particular choice of the user for a method or another, all the algorithm need the arrangement of an equation for pressure that can be derived by proper combination of both continuity and momentum transport equations.

This can be done by taking the divergence of the momentum equation and using the continuity equation to simplify the resulting expression, ending up with a Poisson equation for the pressure.

The application of the divergence operator to the momentum Equation (2.22) leads to:

$$\nabla \cdot (\nabla p) = -\nabla \cdot \left[\nabla(\rho \mathbf{U} \mathbf{U} \mathbf{S}) - \rho \mathbf{b} + \frac{\partial(\rho \mathbf{U})}{\partial t} \right] \quad (2.73)$$

which, in case of incompressibility can be further simplified by leveraging the divergence-free constraint:

$$\nabla \cdot (\nabla p) = -\nabla \cdot [\nabla(\rho \mathbf{U} \mathbf{U})] \quad (2.74)$$

The pressure equation so obtained can than be solved using one of the numerical methods discussed in Paragraph 2.5.5. Moreover, it must be noted that Equation (2.74) contains in its RHS derivatives of terms coming from the momentum equation. In order to ensure continuity, these terms must be discretized in a consistent manner respect to the terms they are derived from.

For the steady-state RANS analyses of Chapter 4 the SIMPLEC algorithm has been adopted, while for all the LES simulations presented in this

work it has been chosen the transient PISO algorithm (reported in Figure 2.4), with the exception of the virtual flow bench analyses.

For these latter, an hybrid transient scheme PISO-SIMPLEC, namely PIMPLEC, has been employed. Such scheme, despite the standard PISO, doesn't need the adoption of timesteps strictly guaranteeing $Co < 1$ and, thanks to its SIMPLEC features inheritance, is more robust on complex grids.

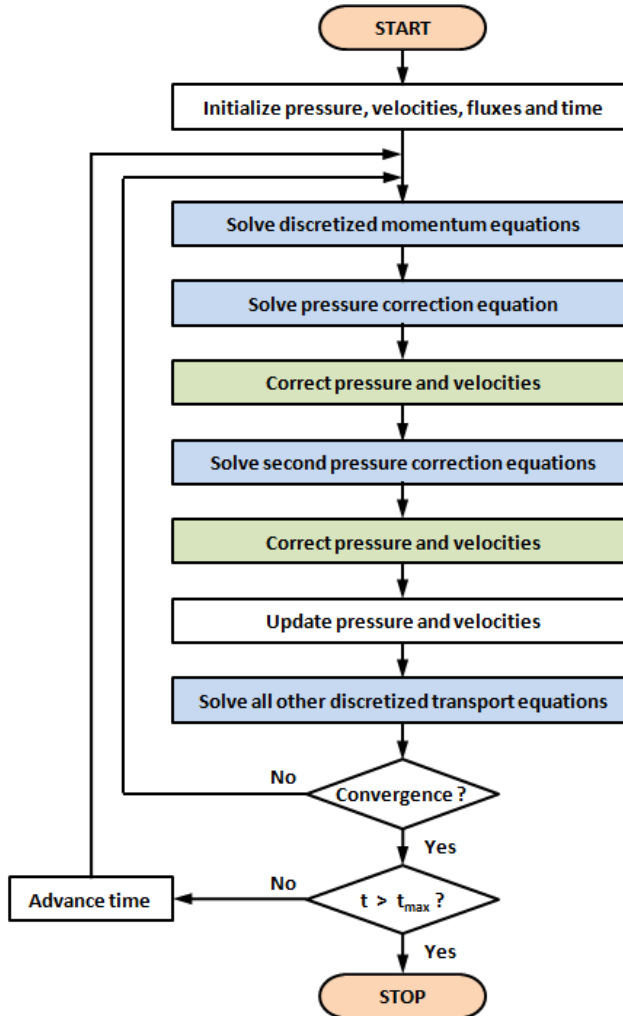


Figure 2.4: Flowchart of the PISO algorithm. Adapted from [10].

2.5.7 A few words about *OpenFOAM*[®]

OpenFOAM[®], which is the acronym of *Open-Source Field Operation and Manipulation*, is an open-source CFD package written in the C++ programming language developed primarily by *CFD Direct Ltd.*, on behalf of the *OpenFOAM Foundation*, and it is released as free and open-source software under the GNU General Public License [38]. *OpenFOAM*[®], (originally just *FOAM* [39]) has been created by H. Weller in the late 1980s at Imperial College (London) while trying to develop a flexible simulation toolbox with greater flexibility than the one allowed by FORTRAN, which was, at that time, the standard programming language for CFD codes. C++ has then been the natural choice, due to its flexibility and its object-oriented features.

Object-orientation allows to easily maintain and modify existing code since new objects can be created with minor changes of existing ones. For instance, the mechanism of dynamic linking allows to define a class and compile it into a shared object library. When new code using such class will undergo compilation, the class itself will not be recompiled, but rather the class will be recalled at runtime from the new code.

Furthermore, object-orientation has been the key to implement one of the most peculiar features of *OpenFOAM*[®]: its syntax. Tensor operations and partial differential equations can be in fact coded in a way that closely resembles the scientific mathematical language, simplifying the programming of new solvers and libraries [39].

Being open-source, *OpenFOAM*[®] has acquired along the years a remarkable popularity in the CFD codes arena. The community of users and developers has grown tremendously and today *OpenFOAM*[®] is with no doubt the most widely adopted open-source CFD code in both academia and industry. Community-driven efforts in code customization and extension have allowed *OpenFOAM*[®] to become an advanced CFD package capable to handle complex flows such as chemically reacting flows, particle flows, flows with moving domains and conjugated heat transfer, just to cite some.

Several of the algorithms and the discretization procedures presented so far are available in *OpenFOAM*[®], which operates under the FV framework.

The adoption of *OpenFOAM*[®] for the present work was not only due to the possibility to extend the code as needed, but also to allow deeper insights into the procedures and algorithms used. Finally, the absence of any licensing cost has been a significant advantage, in particular when dealing with resource-demanding LES simulations.

2.6 Numerical dissipation and dispersion

It is well known by literature that numerical schemes tend to introduce numerical dissipation and dispersion on the flow field solution ([8], [30]). In explicit LES analyses the dissipation of kinetic energy is provided directly as viscous dissipation or through SGS dissipation, while the numerical dissipation of the code should be kept as low as possible (ideally zero). Furthermore, the importance of properly resolving the time evolution of the turbulent flow structures leads to the need of minimize numerical dispersion as well. It is therefore interesting to evaluate the attitude of *OpenFOAM*[®] towards both kinds of the aforementioned numerical errors to estimate its suitability for LES computations.

In order to accomplish such evaluation, a specific test case will be used. This consists in a vortex convected by a uniform inviscid flow in a 2D periodic domain, as proposed by Yee et al. in [40]. Supposing that there are no numerical dissipation or dispersion errors, after one or more convective times T_0 , the vortex profile remains the same as the initial (analytic) one. Similar tests employing a Taylor-Green vortex have been performed by other Authors ([41], [42] and [43]) as a method to assess code dissipation. To be noted that in the following study the dispersion error has been also evaluated, as done by Brusiani et al. in [44].

The computational domain, reported in Figure 2.5, has been discretized using a 2.5D uniform hexahedral mesh with one cell only along z direction, accordingly to *OpenFOAM*[®] requirements for pseudo-2D analyses. Three different grid densities have been used employing 12, 24 and 48 points along the vortex radial profile, with mesh sizes of 48×48 , 96×96 and 196×196 cells respectively. Given $D_0 = 2\pi m$, the initial diameter of the vortex, the domain is a square with side of $2D_0$. The domain extent has been chosen in order to minimize the effects of the boundary conditions on the solution.

The initial conditions for the fluid-dynamic vortex flow field were:

$$u = -\frac{\beta}{2\pi} y e^{\frac{1-x^2-y^2}{2}} \quad (2.75)$$

$$v = \frac{\beta}{2\pi} x e^{\frac{1-x^2-y^2}{2}} \quad (2.76)$$

$$p = p_0 - \frac{\beta^2}{8\pi^2} e^{(1-x^2-y^2)} \quad (2.77)$$

where $\beta = 5s^{-1}$ is the vortex strength and $p_0 = 101325 Pa$. A constant convective velocity U_0 of $10 m/s$ has been added along the x direction of the profiles in order to reproduce the convective transport process. Symmetries have been used for the top and bottom domain boundaries while periodic conditions have been applied on the other sides, as shown in Figure 2.5.

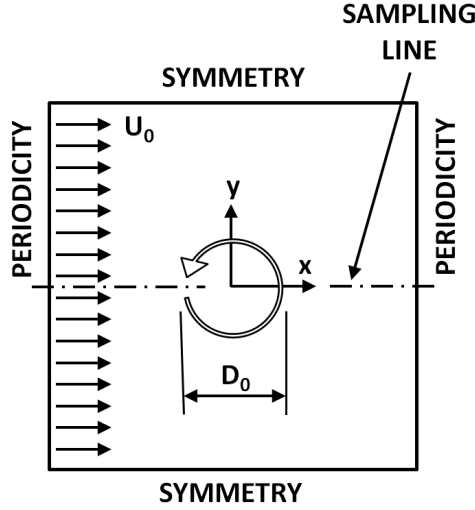


Figure 2.5: Taylor vortex test setup.

The *OpenFOAM*[®] solver used is *potentialTransientFoam*, a solver which has been specifically implemented during the present work for the solution of transient inviscid fluid flows. It is based on a second order backward implicit integration scheme for time advancement and employs a Bounded Central Difference scheme (BCS) for pressure and momentum equations. The PISO algorithm provides the pressure-velocity coupling.

Numerical results are reported in Figures 2.6 - 2.11. Figures 2.6 and 2.7 show the influence of the grid density on the calculation accuracy by comparing the analytic solution of the vortex transport against the numerical solution after respectively 1 and 5 convective times T_0 . All the three cases have been run with a Courant Number $Co = 0.5$ and the pressure profiles are taken from a sampling line coincident with the x axis. It is evident how decreasing the mesh resolution adversely affects the vortex profile: the pressure peak is progressively reduced and the shift in phase with the exact solution increases. Furthermore spurious oscillations of the pressure and velocity field (here not reported) appear when the coarsest mesh is used. In such case the vortex is quickly smeared out and some drift of the vortex

core from the centerline tend to occur.

In Figures 2.8 and 2.9 is shown the influence of the Co on the solution accuracy for the medium density grid. The figures report the vortex pressure profile taken on the same post-processing topology as done for the mesh influence evaluation. The effects of the time discretization are particularly manifest in Figure 2.9 where both vortex smearing and shift in phase appear with increasing Co .

Finally, Figures 2.10 and 2.11 summarize the results obtained in terms of numerical dissipation and dispersion. In the former plot the decrease of the vortex total kinetic energy is reported and the effect of the grid resolution on the energy conservation is clearly shown. Moreover the plot suggests that the influence of the space discretization on the vortex kinetic energy preservation is greater than the influence of the time discretization. Such a result is confirmed also by [42] in which the Authors found a significant improvement in the numerical accuracy with the grid resolution. The dispersion error is evaluated by considering the pressure peak vortex drift with respect to the analytic solution, normalized by the vortex original diameter D_0 . In this case both space and time discretization seem to affect the numerical accuracy with a greater influence of the former parameter.

The outcomes of this test give some guidelines on the numerical setup of LES analyses with *OpenFOAM*[®], suggesting to pay attention to the levels of grid refinement and, possibly, running the simulation with $Co < 0.5$. All the tests presented in Chapter 4 have been carried out with a maximum Co between 0.3 and 0.5 with some exceptions for the finer grids in which the threshold has been raised to 0.65 to reduce the computing time. Providing specific recommendations for space discretization is not trivial, since it strongly depends on the dimension of the smaller vortical structures that one needs/wants to resolve. Moreover, in case of ICE simulations the characteristic length scale varies with operation type (motored/fired), engine regime and engine phase (see [45], [46], [47], [48], [49] or [50] for more details on this issue). Once the characteristic dimensions of the structures intended to be resolved are known, the aforementioned findings could provide an indication of the amount of points needed for proper resolution, and, from this, an estimate of the grid spacing.

2.6. Numerical dissipation and dispersion

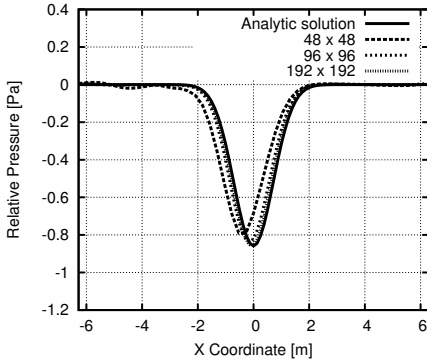


Figure 2.6: Mesh influence - Relative pressure profile after one convective time.

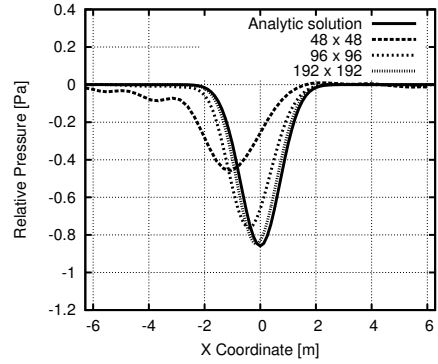


Figure 2.7: Mesh influence - Relative pressure profile after five convective times.

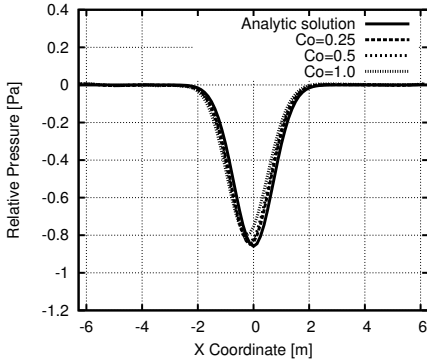


Figure 2.8: Courant number influence - Relative pressure profile after one convective time.

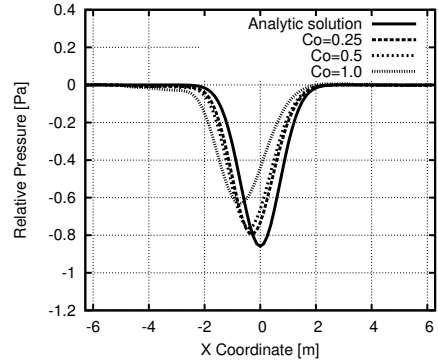


Figure 2.9: Courant number influence - Relative pressure profile after five convective times.

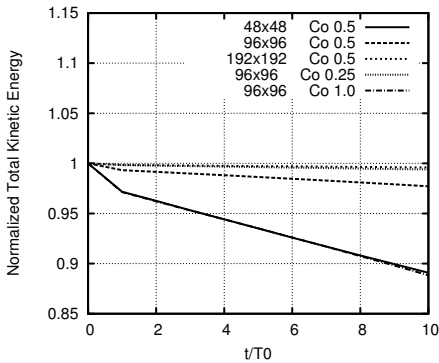


Figure 2.10: Numerical dissipation as a function of the convective time.

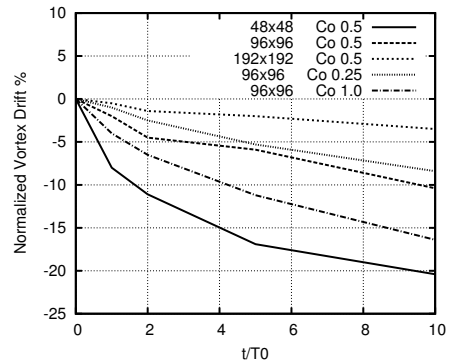


Figure 2.11: Numerical dispersion as a function of the convective time.

CHAPTER 3

LES Quality Assessment

3.1 General remarks on LES analysis quality assessment

In the last decade the number of works dealing with LES of ICE flows constantly increased due to the potential this modeling approach has to offer within this field as shown in Figure 3.1. Despite that, most of the scientific publications lack the resolution quality assessment of the simulations with some noticeable exceptions such as [51], [52], [53] and [54].

The estimation of the quality of LES results is not trivial. A reason for this is that they are very sensitive to both grid resolution and the numerical method employed. Furthermore, the grid resolution affects not only the numerical discretization error but also the sub-grid scale model contribution. Since the basic idea behind LES is to resolve all the energy carrying larger eddies, while the dissipative smaller eddies are modeled, the quality estimation try somehow to evaluate the extent and completeness of such resolution.

To be noted that quality indicators should be considered tools for verification rather than validation. They are in fact computed using merely the quantities returned by the simulations without requiring any comparison with DNS or experimental data. While validation can be performed

using solution statistics of different orders (mean, rms, etc.) with the due attention to the pitfalls of such practice [55] or using theoretical considerations (e.g. energy spectra), quality parameters allow an evaluation of the grid resolution adequacy and the sub-grid scale modeling characteristics.

Among the LES quality parameters, it is possible to distinguish between single-grid estimators and multi-grid estimators. While for the former ones a single LES analysis is sufficient to retrieve the values of interest, for the latter, at least two LES calculations have to be performed. Such additional computational effort required make them not particularly popular for ICE flows applications and their use is almost confined to theoretical studies [56].

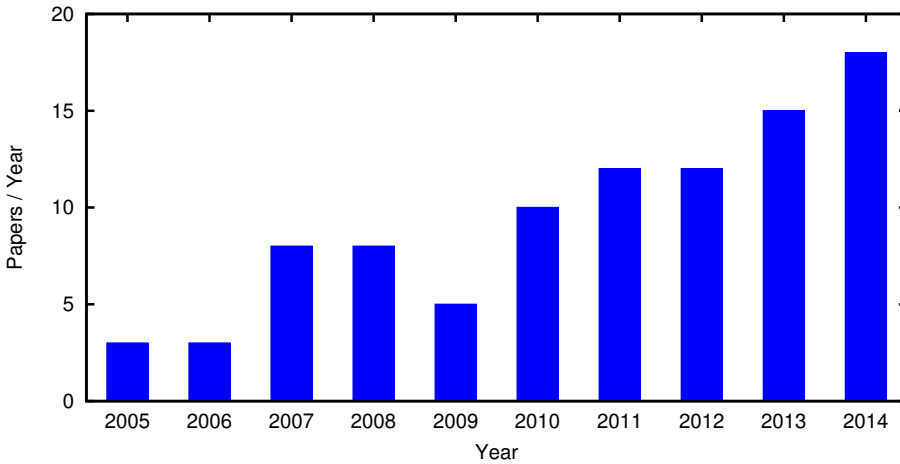


Figure 3.1: Number of ICE related published papers per-year which have the words “LES” in title, abstract, or keyword list. Rearranged from [57].

In the following paragraphs, several LES quality assessment parameters are presented and discussed. It must be noted however that these represent only a subset of a larger number of resolution tests available.

3.2 LES quality estimators

3.2.1 Fraction of resolved kinetic energy - M

The amount of turbulent kinetic energy resolved by a simulation is a key aspect for evaluating LES quality. In [58] Pope defines the parameter $M(\mathbf{x}, t)$ as a measure of the turbulent resolution. Such parameter weights the amount of turbulent kinetic energy of the residual motion with respect to the total turbulent kinetic energy of the flow:

$$M(\mathbf{x}, t) = \frac{k_{sgs}(\mathbf{x}, t)}{k_{sgs}(\mathbf{x}, t) + k_{res}(\mathbf{x}, t)} \quad (3.1)$$

where $k_{res}(\mathbf{x}, t)$ is the resolved turbulent kinetic energy and $k_{sgs}(\mathbf{x}, t)$ is the residual motion turbulent kinetic energy. The value of M is bounded between 0 and 1. When $M = 1$ a RANS simulation is retrieved, while for M approaching zero, the simulation tends to a DNS. The smaller the parameter M is, the higher the resolution of the turbulent motions is. Pope suggests a threshold value of $M < 0.2$ as a reasonable resolution limit for LES.

It must be noted, however, that the parameter M alone is not sufficient to evaluate the accuracy of the simulation with respect to the actual flow since M does not account for the fraction of turbulent kinetic energy lost due to numerical dissipation. Furthermore, a high value of M at a certain point in the domain doesn't necessarily mean that in such location the grid needs refinement because the fraction of turbulent kinetic energy could be affected by the filter size in upstream locations and / or earlier in time.

A wall-bounded flow can be fully resolved only if the near-wall region is discretized with a properly refined mesh in all three directions, given the three-dimensional nature of turbulence. In ICE flows such a requirement is hard to fulfill due to the huge computational effort needed. In consideration of this and trying to address feasible modeling strategies, all the turbulence models proposed in the present work make use of wall functions, if not otherwise stated. As a consequence, in these regions the parameter M will locally fall below the aforementioned threshold.

3.2.2 Length Scale Resolution - LSR

The Length Scale Resolution (*LSR*) has been introduced by Brusiani et al. in [44] attempting to estimate the attitude of the grid to properly resolve the scales of fluid motion up to the inertial subrange. It is defined as the ratio between the actual filter size Δ and (an estimate of) the inertial subrange lower limit l_{DI} :

$$LSR(\mathbf{x}, t) = \frac{\Delta}{l_{DI}(\mathbf{x}, t)} \quad (3.2)$$

where $\Delta = \sqrt[3]{\Delta x \Delta y \Delta z}$ and $l_{DI} \approx 60\eta$ with η being the Kolmogorov microscale length [6]. To be noted that such an estimate of l_{DI} is based on scaling assumptions of isotropic turbulence under equilibrium conditions. In the present work the above parameters are estimated as follows:

$$\Delta = \sqrt[3]{\Delta x \Delta y \Delta z} \quad (3.3)$$

$$\eta(\mathbf{x}, t) = \nu^{3/4} \varepsilon(\mathbf{x}, t)^{-1/4} \quad (3.4)$$

$$\varepsilon(\mathbf{x}, t) \approx \frac{C_\varepsilon k(\mathbf{x}, t)^{3/2}}{\Delta} \quad (3.5)$$

The LSR parameter is therefore defined as the ratio between the actual filter dimension and the characteristic length of the inertial range motions. Such parameter correlates the filter dimension Δ to the fraction of resolved energy spectrum and is therefore useful to evaluate the optimality of the computational domain discretization, identifying regions that could need refinement. While having $LSR = 1$ means complete length scale resolution up to the dissipative range, accordingly to the Authors' findings [44], $LSR < 5$ is a reasonable threshold for an adequate energy resolution level at an affordable computational cost.

3.2.3 Viscosity ratio - ν_{SGS}/ν

The ratio between the sub-grid viscosity ν_{SGS} and the molecular viscosity ν is representative of the degree of modeling in a LES simulation:

$$s' = \frac{\nu_{SGS}}{\nu} \quad (3.6)$$

Several Authors prefer to weight the amount of turbulent viscosity with respect to the effective viscosity defining the following sub-grid activity parameter:

$$s'' = \frac{\nu_{SGS}}{\nu_{SGS} + \nu} \quad (3.7)$$

However, as Celik pointed out in [56], since $\nu_{SGS} \gg \nu$ the s'' parameter is often pretty close to unity and not really sensitive to grid resolution. For such reason in the present work the s' formulation has been used. Values of $s' < 10$ are considered sufficient to obtain an accurate LES as shown by Durbin in [59] or by Poinso in [60].

3.2.4 Sub-grid Scale Activity - s

In one of the first attempts to define LES quality parameters, Geurts has suggested an estimator, s , based on the subgrid activity [61]. The parameter is defined as:

$$s = \frac{\langle \varepsilon_t \rangle}{\langle \varepsilon_t \rangle + \langle \varepsilon_\mu \rangle} \quad (3.8)$$

where $\langle \rangle$ indicates an average operation over the whole computational domain, ε_t is the turbulent dissipation and ε_μ is the molecular dissipation. By definition, $s = 0$ corresponds to a DNS, while $s = 1$ corresponds to a LES at infinite Reynolds number. As Celik demonstrated in [56], Equation (3.8) can be rearranged in a form similar to (3.6), inheriting the relative drawbacks. For such reason this parameter has not been used in the present work.

3.2.5 Two grid estimator - LES_{IQk}

The LES_{IQk} parameter has been proposed by Celik [56] who defined it as the ratio between the resolved turbulent kinetic energy k_{res} and the total turbulent kinetic energy $k_{tot} = k_{res} + k_{SGS} + k_{num}$:

$$LES_{IQk} = \frac{k_{res}}{k_{tot}} = \frac{k_{res}}{k_{res} + k_{SGS} + k_{num}} \quad (3.9)$$

When $LES_{IQk} = 0$ a RANS simulation is retrieved, while the LES simulation approaches a DNS when LES_{IQk} tends to unity. By assuming that the effective sub-grid turbulent kinetic energy $k_{eff,SGS} = k_{SGS} + k_{num}$ scales with the mesh size, the Author reformulates Equation (3.9) as:

$$LES_{IQk} = \frac{k_{res}}{k_{res} + a_k h^p} \quad (3.10)$$

where a_k is a coefficient to be determined, p is the formal order of accuracy of the numerical scheme and h is the grid size. To be noted that the formulation of this quality indicator is well suited for implicitly filtered LES, where a modification of h is directly reflected onto the filter size Δ , and, from this, onto both the numerical and SGS behaviors jointly. Denoting with superscript 1 the finer grid and with superscript 2 the coarser one, Equation (3.10) can be reformulated as:

$$LES_{IQk} = \frac{k_{res}}{k_{res} + \frac{k_{res}^2 + k_{res}^1}{\alpha^p - 1} \left(\frac{h}{h_2} \right)^p} \quad (3.11)$$

where $\alpha = h_1/h_2 > 1$ is the grid refinement ratio. Equation (3.11) can be then rearranged for both grid sizes. Here the expression for the finer grid is reported:

$$LES_{IQk}^{fine} = \frac{1}{1 + \left(1 - \frac{k_{res}^1}{k_{res}^2}(\alpha^p - 1)^{-1}\right)} \quad (3.12)$$

In this approach the inherent difficulty of evaluating the code numerical dissipation is hidden into the parameter a_k which is computed through a Richardson's extrapolation. The extrapolation is performed with results obtained on two grids, similar in structure but with different refinement levels. Due to this last aspect, the LES_{IQk} parameter belongs to the class of two-grid estimators. The procedure is certainly computationally expensive and can lead to misleading results in practical applications as found by Di Mare [54], and this explains its scarce adoption, even in academic works. Despite these issues, this estimator has been used in the present work in order to test its capabilities in evaluating the code's numerical dissipation.

CHAPTER 4

Basic Test Cases

4.1 Dellenback abrupt expansion

4.1.1 Test case and computational setup

The implemented SGS models discussed in Section 2.4 have been firstly validated on a test case related to a swirling flow through a sudden expansion that has been experimentally investigated by Dellenback et al. [62]. The swirl number, defined as:

$$S = \frac{\int_0^R U_\theta U_z r^2 \, dr}{R \int_0^R U_z^2 r \, dr} \quad (4.1)$$

was approximately 0.98, based on the inlet radius $R = D_0/2$. U_θ and U_z are respectively the time-averaged tangential and axial velocity components. The Reynolds number, based on the inlet diameter D_0 and the inlet bulk velocity was $Re = 30000$. Even if the case may seem a little far from being a flow of interest for engine application, it has several features that can be found in real engine flows, such as swirl, flow detachment and recirculating regions. In fact, the case has already been subject of numerical

investigation by other Authors, amongst which Schlueter [63], Javadi [64] and Piscaglia [65].

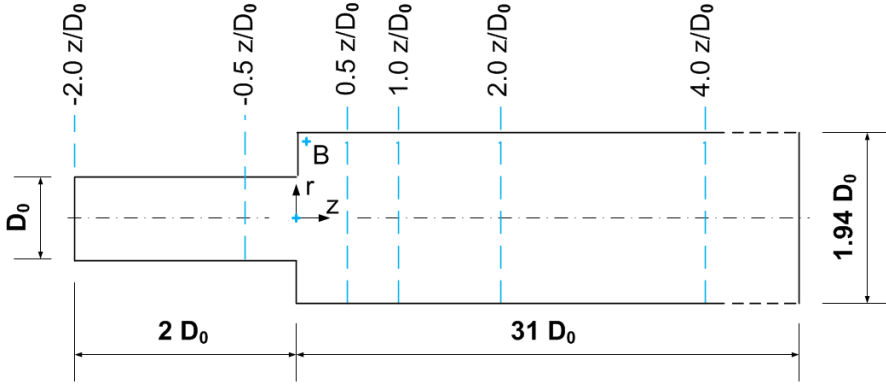


Figure 4.1: Geometry of the Dellenback test case and probe locations.

The geometry of the case is reported in Figure 4.1. The inlet duct has a diameter $D_0 = 0.0508 \text{ m}$ and a length of $31D_0$ with the expansion located $15D_0$ downstream the swirl generator. The duct diameter after the expansion is $D_1 = 0.0985 \text{ m}$, giving an expansion ratio of 1:1.94. The experiments have been performed using water and the swirl has been generated supplying a portion of the flow through tangential slots. Axial and tangential velocity components of mean and rms components have been measured using Laser Doppler Anemometry (LDA) on different radial stations located through the duct midsection as shown in Figure 4.1.

OpenFOAM® v. 2.3.0 has been used for the simulations. The temporal derivatives have been discretized using a second-order backward differencing scheme. For momentum convection and diffusion terms a second-order central difference scheme has been employed. The pressure-velocity coupling has been performed using the PISO algorithm.

Different types of grids have been used for the simulations:

- Fully hexahedral coarse grid - 0.7 M cells
- Fully hexahedral medium grid - 1.5 M cells
- Fully hexahedral fine grid - 6.0 M cells
- Fully tetrahedral medium grid - 1.5 M cells

The hexahedral grids are all block-structured with a slightly higher resolution near the expansion. No near-wall mesh refinement has been explicitly applied and wall functions have been used to handle such region. This

approach is justified by the needs of ICE flow computations where a full near-wall resolution is extremely difficult to achieve, at least in real-world cases. The y^+ in the expansion region walls ranges from 15 to 50 depending on the grid.

Time discretization has been performed ensuring a maximum $Co < 0.5$ in light of the considerations made in Section 2.6, employing timesteps in the range of 10^{-4} - 10^{-5} s.

Both the solver and the turbulence models adopt an incompressible formulation. Even if compressibility effects in ICE flows are important, the objective of this part of the work has been the evaluation of the implemented SGS models, whose performance does not depend on the compressibility of the flow.

At the pipe inlet mean velocities profiles based on experimental data have been mapped onto the domain. No turbulent fluctuations have been explicitly set. The choice of using a "pseudo-laminar" inlet boundary condition has been motivated by the high level of swirl, capable of inducing a fast transition to turbulence, as shown in [63]. At the expansion outlet a homogeneous zero-gradient boundary condition has been set for velocity and a fixed value of zero relative pressure has been set. Finally, a no-slip condition has been applied at walls.

4.1.2 Results - Velocity profiles

A first evaluation of the analysis results is made by comparing the velocity mean and RMS profiles at different locations with the corresponding experimental data. The profiles have been averaged over 12 mean flow advection times and circumferentially averaged.

Figures 4.2 - 4.5 show the velocity profiles and the corresponding turbulent fluctuation levels obtained with the different SGS models on the coarse grid. Looking at the profiles at $z/D = -0.5$, it can be noticed that even if employing a pseudo-laminar inlet condition the mean profiles are correctly predicted and the turbulent fluctuations have grown pretty close to the experimental ones thanks to the high swirl level, as explained in the previous section. The evaluation of the profiles downstream the expansion reveals an overall good agreement with the measurements with some deficiencies in both the mean and RMS axial profiles close to the pipe axis. The WALE and the Sigma models appear to perform slightly better than the other models. The dynamic Smagorinsky seems to be the less accurate model, in particular close to the walls. This is not surprising since it is known by literature to over-predict viscosity in these regions.

Figures 4.6 and 4.7 display the velocity profiles obtained with the Sigma SGS models for the three hexahedral grids. Both the coarse and the medium density grids seem to resolve with reasonable accuracy the average profiles and RMS fluctuations with a lower agreement at the pipe axis. By contrast, the predictions made with the fine grid denote poor agreement with the experimental profiles. This is particularly evident for the RMS fluctuations in the pipe bulk region. The other SGS model profiles (here not reported for the sake of brevity) exhibit a very similar behavior. A possible reason for this could be an improper tuning of the models constants (C_{dS} , C_W , C_{dW} and C_σ). In particular, too low values of such constants could result in an overestimation of the turbulent fluctuations with increasing mesh density, as explained by Piscaglia et al. in [52] facing a similar issue.

The effect of employing grids with different types of element is evaluated through Figures 4.8 and 4.9 which display the velocity profiles obtained with the Sigma SGS model on the medium density hexahedral grid and on a uniform tetrahedral grid. The grids have approximately the same number of elements. Both the mean and the RMS profiles obtained on the tetrahedral grid are in good agreement with the experimental data, with an accuracy comparable to the hexahedral mesh. This is not surprising since the flow past the expansion, where the sampling stations are located, is generally not instantaneously aligned with the grid elements, being them hexahedra or tetrahedra and this is cause of numerical diffusion adversely affecting the solution accuracy as reported by Brusiani in [51].

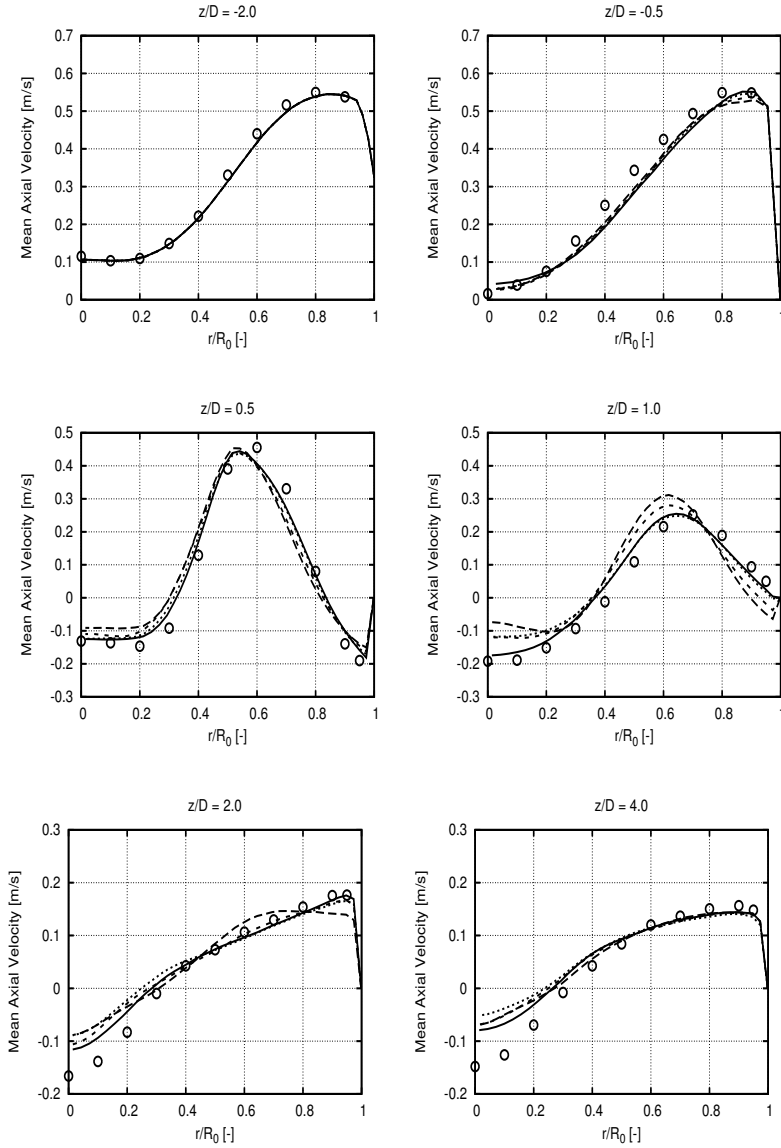


Figure 4.2: Mean axial velocity profiles at different locations for the coarse mesh. Legend: \circ experiment, — — dynamic Smagorinsky, - - - dynamic WALE — WALE, Sigma.

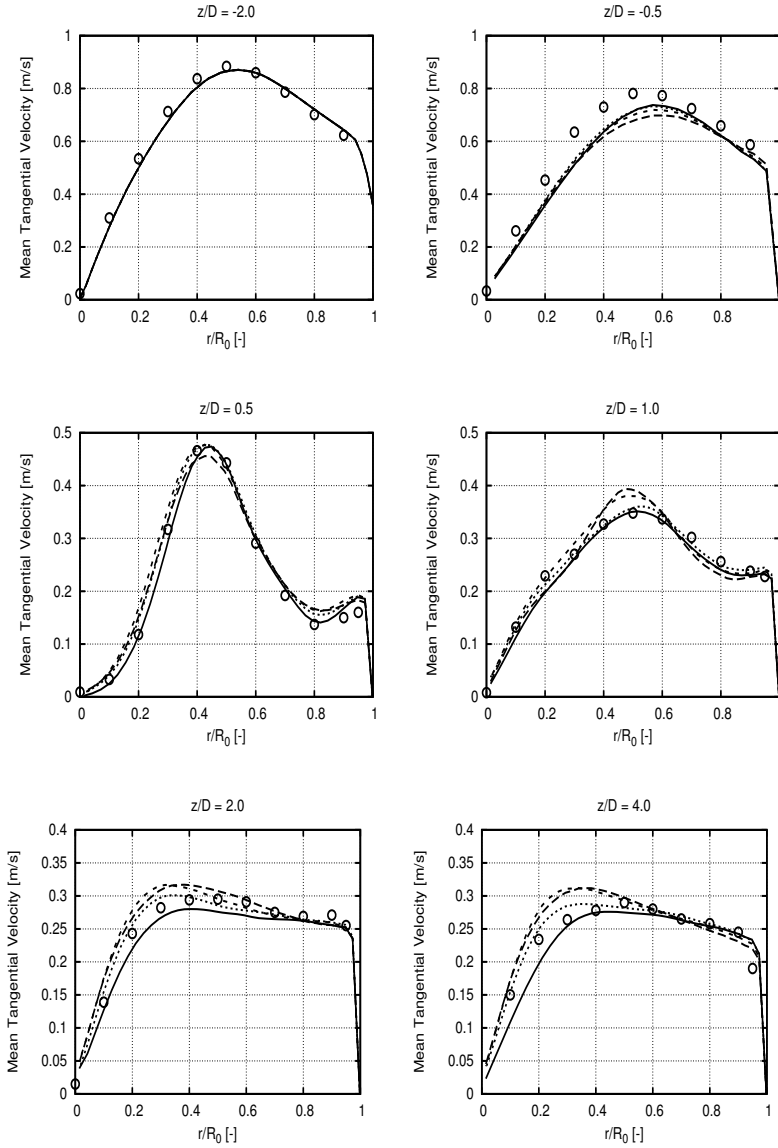


Figure 4.3: Mean tangential velocity profiles at different locations for the coarse mesh.
 Legend: \circ experiment, $--$ dynamic Smagorinsky, $---$ dynamic WALE $—$ WALE, $- \cdot - \cdot$ Sigma.

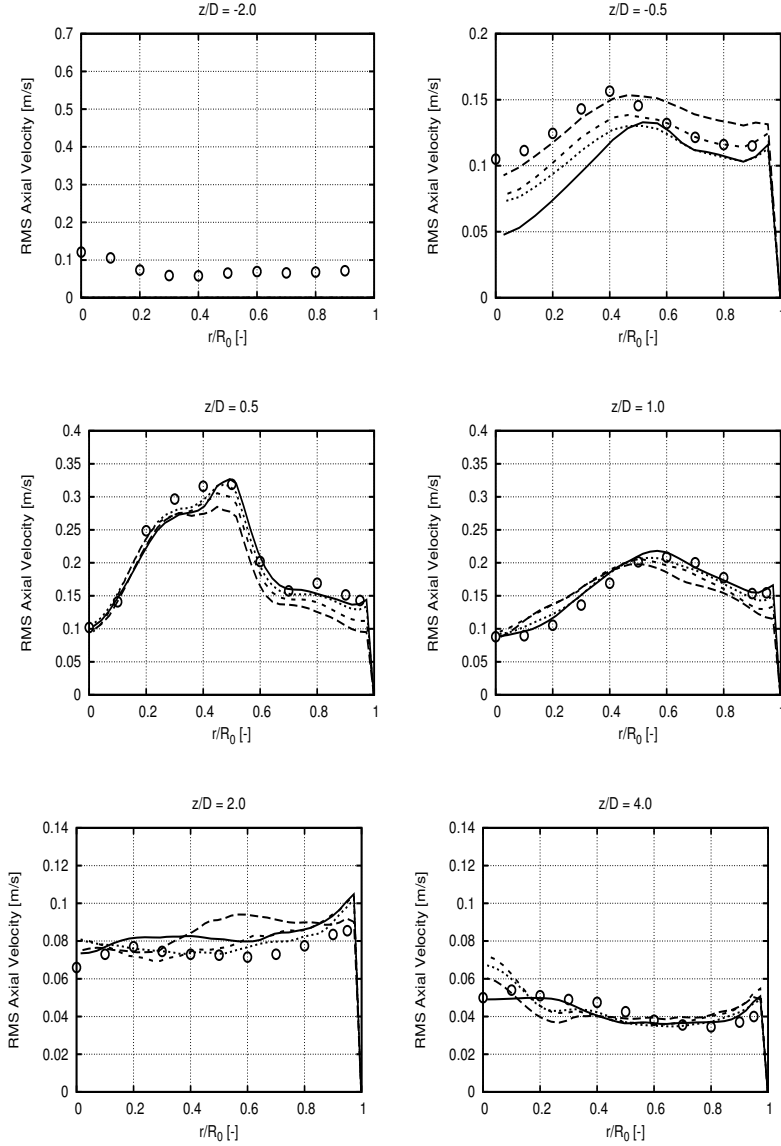


Figure 4.4: RMS profiles of axial velocity fluctuations at different locations for the coarse mesh. Legend: \circ experiment, — dynamic Smagorinsky, - - - dynamic WALE, — WALE, Sigma.

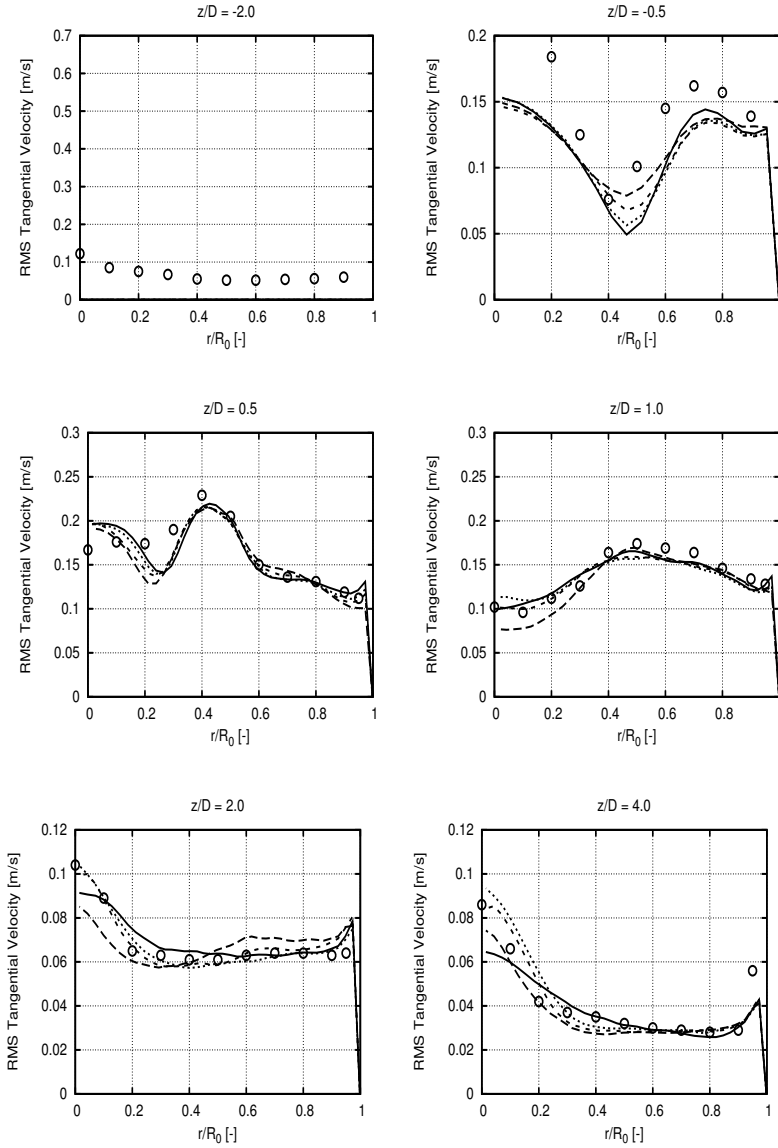


Figure 4.5: RMS profiles of tangential velocity fluctuations at different locations for the coarse mesh. Legend: \circ experiment, $---$ dynamic Smagorinsky, $- - -$ dynamic WALE, $—$ WALE, \cdots Sigma.

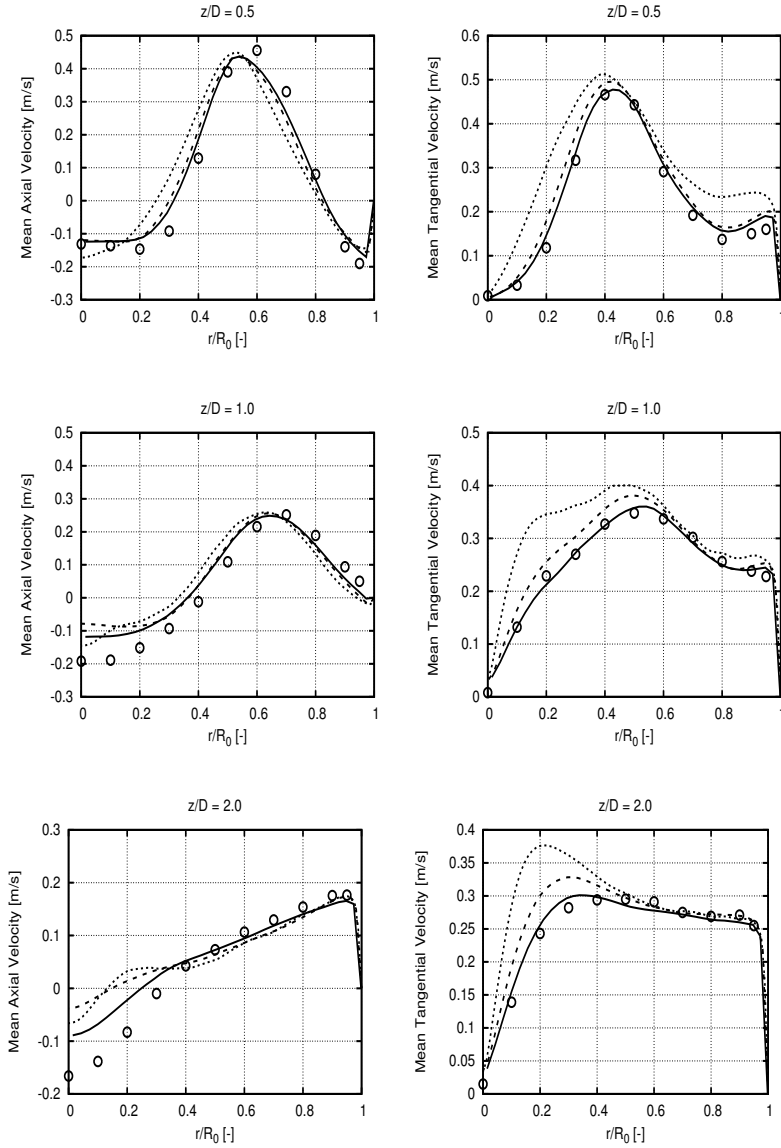


Figure 4.6: Mean velocity profiles for different grid densities (Sigma model). Legend: \circ experiment, $--$ dynamic Smagorinsky, $- \cdot -$ dynamic WALE, $—$ WALE, \dots Sigma.

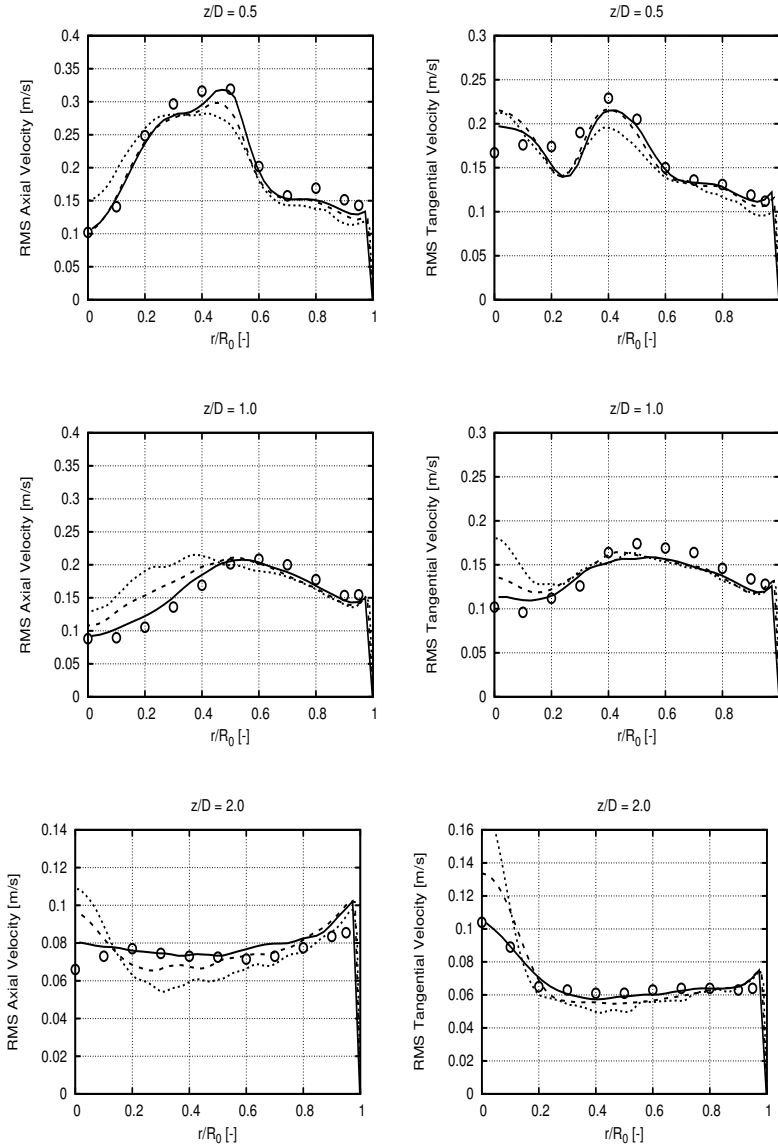


Figure 4.7: RMS profiles of velocity fluctuations for different grid densities (Σ model).
 Legend: \circ experiment, $- -$ dynamic Smagorinsky, $- \cdot -$ dynamic WALE, \cdots Σ .

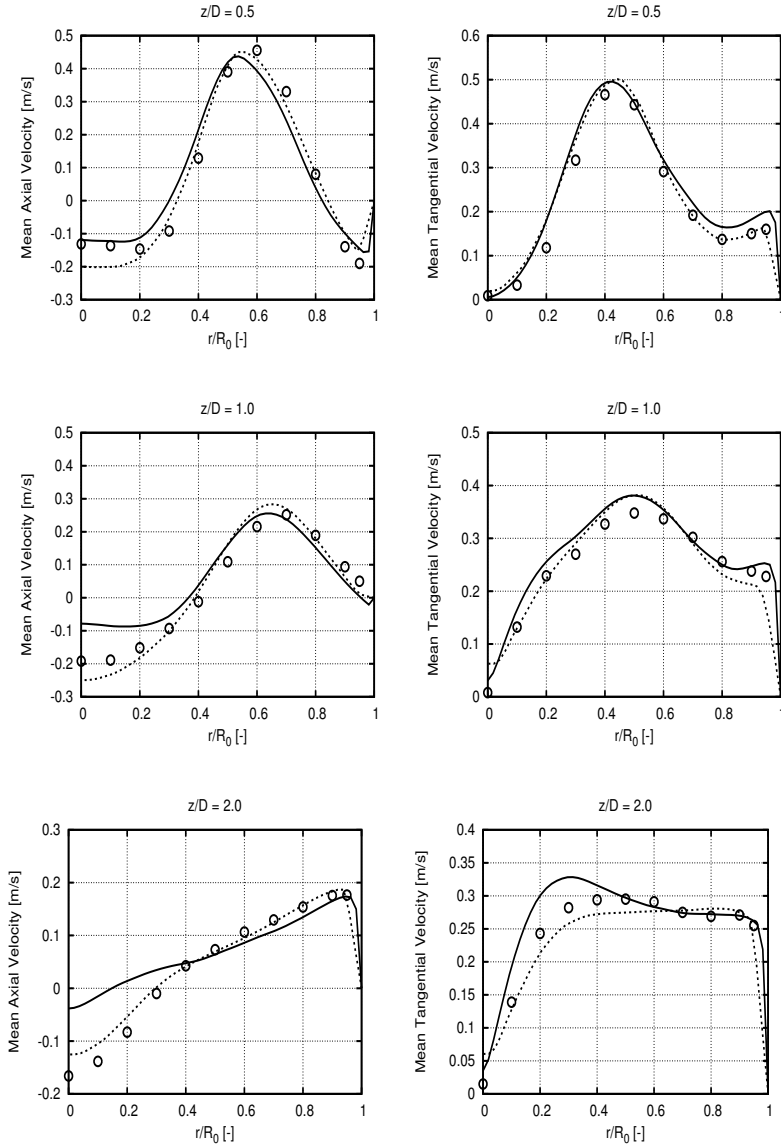


Figure 4.8: Mean velocity profiles for different grid element types (Sigma model - medium density grid). Legend: \circ experiment, — hexa, tetra.

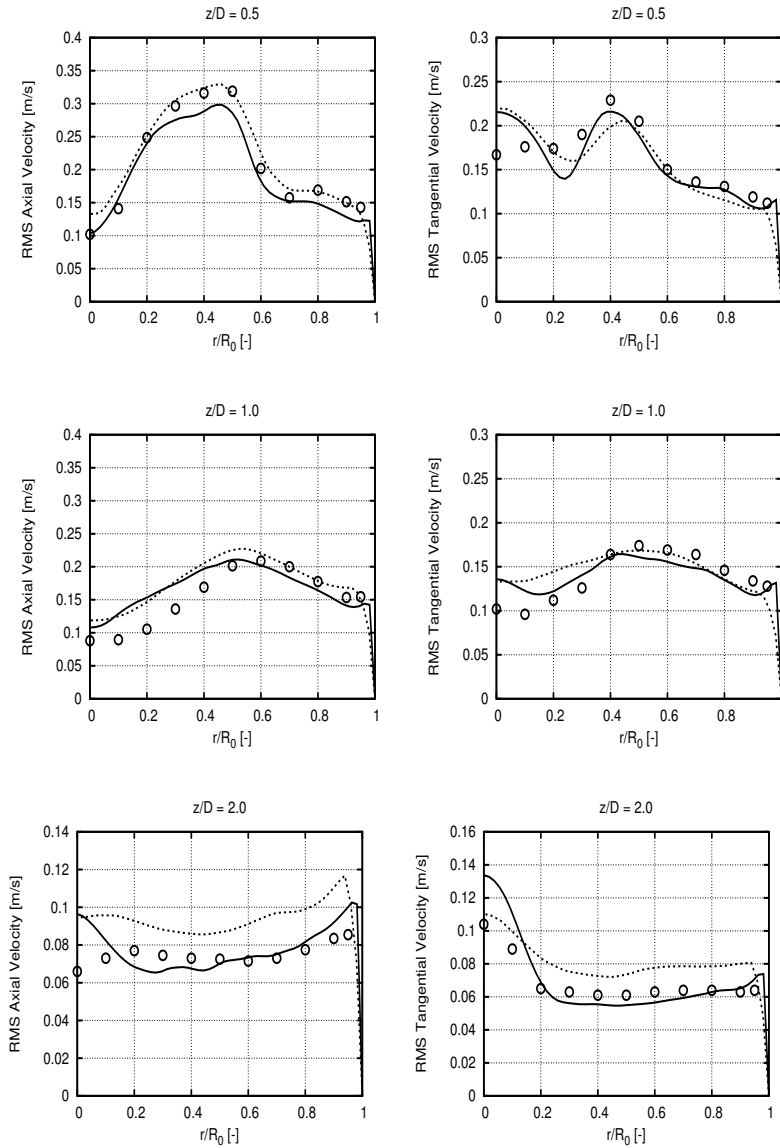


Figure 4.9: RMS profiles of velocity fluctuations for different grid element types (Sigma model - medium density grid). Legend: \circ experiment, — hexa, tetra.

4.1.3 Results - Quality assessment

The quality of LES analyses has been assessed through the parameters discussed in Chapter 3.

Figure 4.10 presents the M parameter comparison for the three hexahedral grids. It appears clearly how a refinement of the grid leads to an increase of the fraction of resolved turbulent kinetic energy. The resolution lowers downstream the expansion due to the slight coarsening of grid and close to walls. In this latter region the M parameter is always higher than the threshold due to the insufficient resolution obtained by using a wall modeling approach and this was somehow expected.

Figure 4.11 shows the LSR parameter contour for the aforementioned grids. The scale resolution improves with grid refinement and the LSR values for the bulk regions are always smaller than 3 with some exception for the coarse grid. Conversely, the near wall region suffers a lack of resolution due to the near wall modeling just mentioned above.

Viscosity ratio contours are proposed in Figure 4.12 where the contribution of the sub-grid scales visibly lowers with grid refinement. High levels of turbulent viscosity tend to remain right behind the expansion close to walls.

Figure 4.14 presents the trends of the two-grid quality estimator LES_{IQk} for three different sampling stations past the expansion. The index appears to drop in almost all cases at the pipe axis and close to wall, probably due to insufficient resolution. Furthermore LES_{IQk} levels increase with grid refinement, even if not in a monotonic way. Such issue has already been tackled by Di Mare, who in [54] explains how an insufficient statistical sample can lead locally to $k_{res}^1 < k_{res}^2$. Finally, comparison maps between the medium and the fine grid LES_{IQk} are presented in Figure 4.13. The parameter distributions, in particular for the medium grid, seem to be quite noisy, maybe due to the aforementioned insufficient statistical sampling. Despite being not so easy to read, both maps present lower resolution levels close to the walls and at the pipe axis, following the trends of the other estimators.

The trends of the quality estimators here presented indicate that the finer the grid is, the greater the level of resolution provided is. Despite this, the overall quality of the analyses, evaluated in terms of agreement between the simulations and the experiments, tends to diminish when increasing the grid density. Other aspects, such as the grid quality and the SGS models constants, could affect such agreement. Furthermore, all the estimators returned qualitatively comparable results with differences lying in the es-

timator range of values and sensitivity. As mentioned in Chapter 3, these LES quality estimators cannot be used as tools for validation of simulations, even if they are all capable of characterizing the flow field in terms of grid resolution adequacy. They can be therefore usefully employed for comparing grids obtained with the same discretization procedure and/or to identify potential regions that need to be refined in order to improve the resolution.

4.1.4 Results - Energy spectra

Simulations energy spectra have been evaluated at point B of the domain (see Figure 4.1) which is located close to the walls, right past the expansion. Figure 4.15 shows the effect of the grid density on the resolved kinetic energy spectrum. All the three hexahedral discretization seems to exhibit a $-5/3$ power law inertial subrange behavior but, the denser is the grid, the better the asymptotic trend is followed, as expected. The influence of the element shape on the spectra is analyzed in Figure 4.16 in which it appears evident a faster depauperation of the spectrum energy content when the tetrahedral grid is employed. Finally, Figure 4.17 demonstrates a substantial lack of influence of the SGS model on the spectrum shape.

4.1.5 Results - Pressure SPD

The spectral density of the pressure fluctuations probed at point B are reported in Figures 4.18 and 4.19. The plots demonstrate how a reduction in the grid density or the usage of tetrahedral elements tends to reduce the spectral content resolution at higher frequencies. Nevertheless, all the discretizations exhibit a peak in the spectral content at a frequency of nearly 6 Hz. This frequency is related to the revolution frequency of the helicoidal vortex core as discussed in [66]. Furthermore, the corresponding non-dimensional frequency is a Strouhal number of $St = f U_b / D = 0.985$ in pretty good agreement with the experimental value of $St_{exp} \sim 1$. Even if less pronounced, the same peak is visible in the energy spectra presented in Figures 4.15, 4.16 and 4.17.



Figure 4.10: M parameter for different grid densities (Sigma SGS model): top - coarse grid, middle - medium grid, bottom - fine grid. In red the regions where $M > 20\%$.

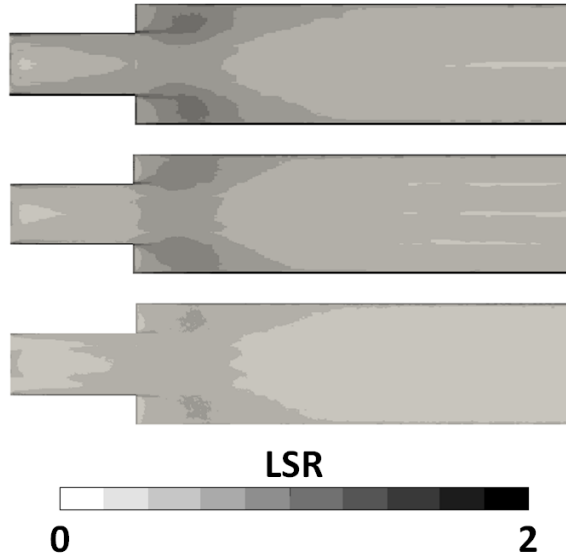


Figure 4.11: LSR parameter for different grid densities (Sigma SGS model): top - coarse grid, middle - medium grid, bottom - fine grid.

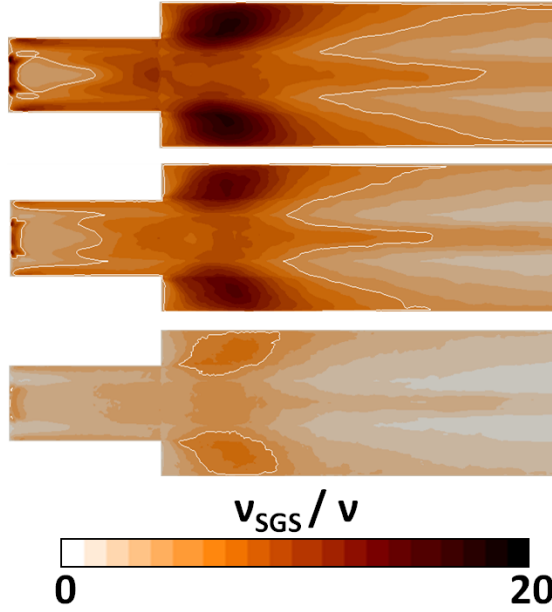


Figure 4.12: ν_{SGS}/ν parameter for different grid densities (Sigma SGS model): top - coarse grid, middle - medium grid, bottom - fine grid. The white contour highlights the isosurface $\nu_{SGS}/\nu = 5$.

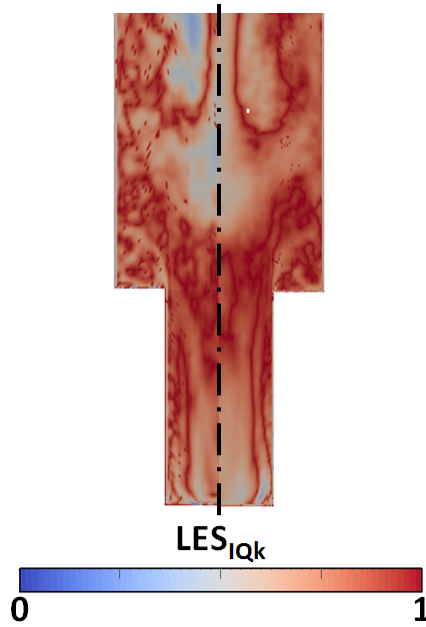


Figure 4.13: LES_{IQk} parameter comparison maps (Sigma SGS model): medium grid (left) and fine grid (right).

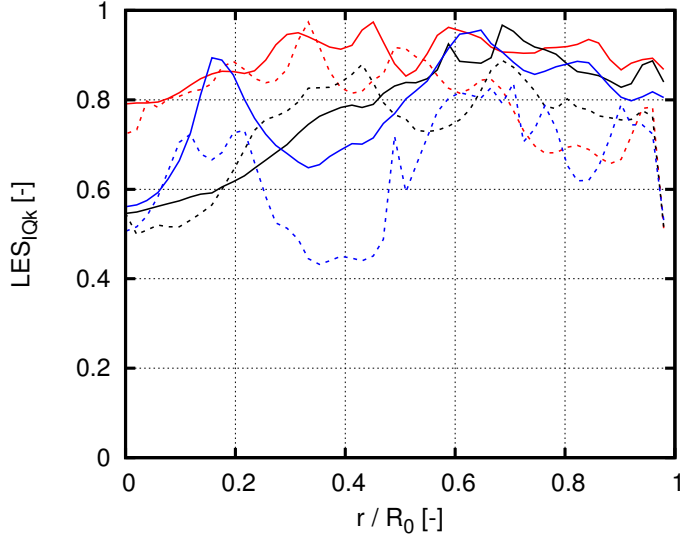


Figure 4.14: LES_{IQk} parameter at different locations (Sigma SGS model): red - $z/D = 0.5$, black - $z/D = 1.0$, blue - $z/D = 2.0$. Dashed line - medium mesh, Solid line - fine mesh.

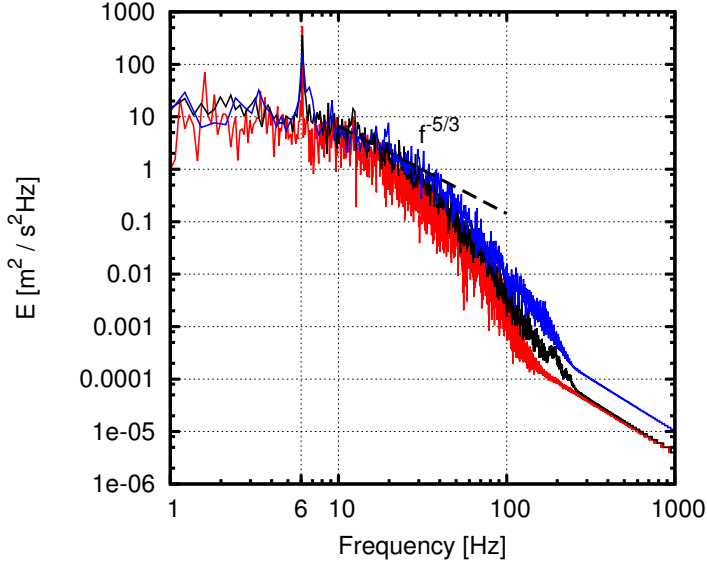


Figure 4.15: Turbulent kinetic energy spectra for different grid densities (Sigma SGS model): coarse grid (red), medium grid (black), fine grid (blue).

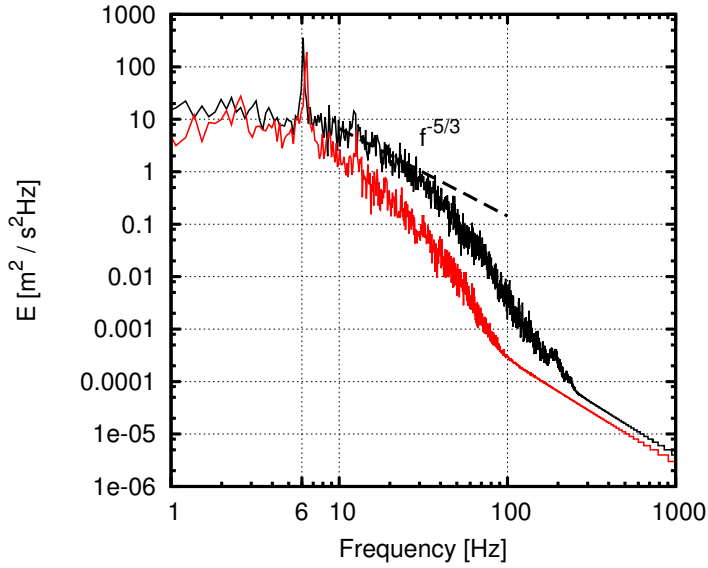


Figure 4.16: Turbulent kinetic energy spectra for different grid element type (Sigma SGS model): tetrahedral grid (red), hexahedral grid (black).

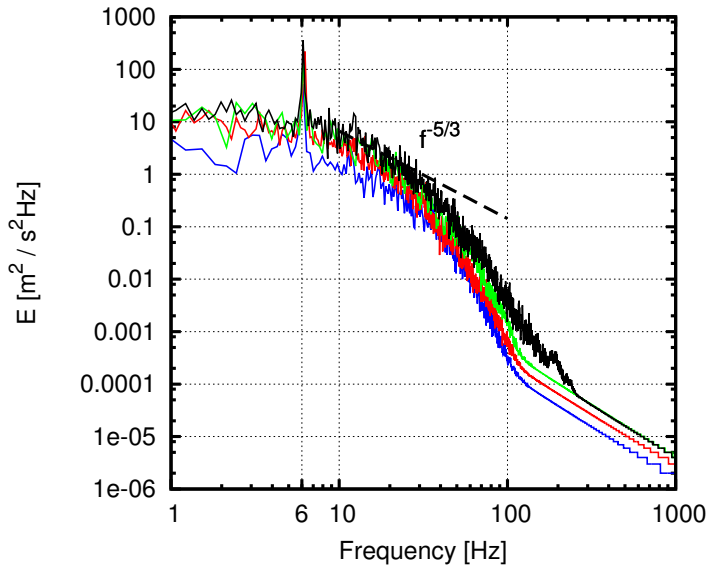


Figure 4.17: Turbulent kinetic energy spectra for different SGS models: dynamic Smagorinsky (blue), WALE (red), dynamic WALE (green), Sigma (black).

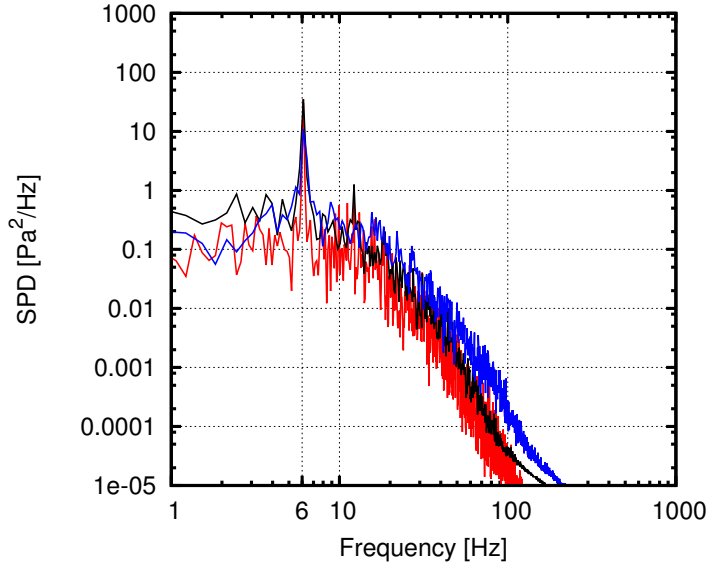


Figure 4.18: Spectral power density of pressure fluctuations at point B (Sigma SGS model): coarse grid (red), medium grid (black), fine grid (blue).

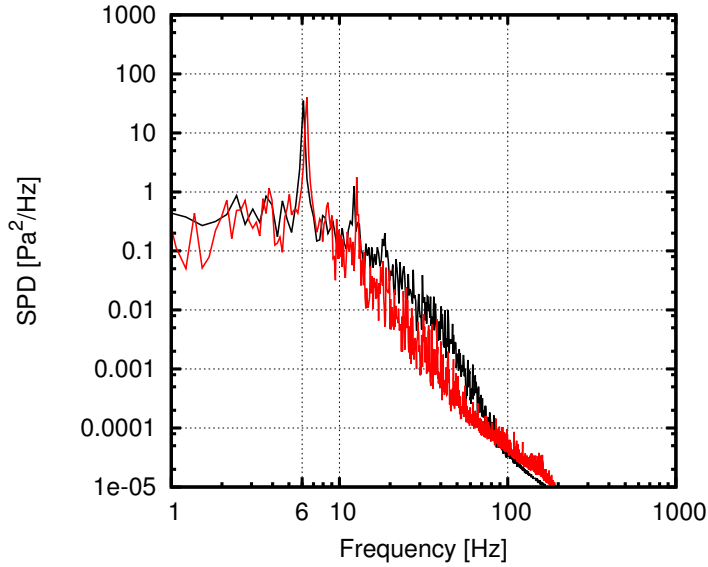


Figure 4.19: Spectral power density of pressure fluctuations at point B for different grid element type (Sigma SGS model): tetrahedral grid (red), hexahedral grid (black).

4.2 Thobois poppet valve

4.2.1 Test case and computational setup

A further validation of the SGS models implemented described in Chapter 2 has been carried out on the flow past a poppet valve that has been experimentally investigated by Thobois [67]. The system is composed by a poppet valve mounted in a flow bench comprising both an intake duct and a discharge duct. The valve is open with a fixed valve lift of 10 *mm* realizing an expansion ratio of approx 3.5. The geometry of the system is shown in Figure 4.20, while its main dimensions are reported in Table 4.1. This case is of special interest for ICE flows, and, in fact, it has been investigated over the years by other Authors, such as Brusiani [68], Piscaglia [52] and Martínez [69].

The experiments have been performed using nitrogen at ambient conditions and no swirl has been applied to the inflow. LDA technique has been used on different radial stations located respectively at 20 *mm* and 70 *mm* downstream the expansion to measure mean axial velocity components as well as RMS of both axial and tangential velocity fluctuations.

OpenFOAM[®] v 2.3.0 has been used for these simulations and the same numerical schemes used for the Dellenback test case have been adopted.

Four different types of grids have been employed for the simulations:

- Fully hexahedral coarse grid, flow aligned – 1.8 M cells
- Fully hexahedral fine grid, flow aligned – 5.6 M cells
- Fully hexahedral coarse grid, orthogonal – 1.8 M cells
- Fully tetrahedral coarse grid – 1.8 M cells

The three hexahedral grids are block-structured and all the grids have been slightly densified close to the valve. The flow aligned grids present cells that are oriented accordingly to jet developing past the valve, while the orthogonal grid has a more usual shape with cells orientation following the topology of the domain. The interest in evaluating the two meshing strategies is due to the eventual resolution differences of the valve jet which, in case of orthogonal grids, crosses the cells at nearly 45 degrees of incidence with potential accuracy degradation. The difference between the two discretization approaches is visualized in Figure 4.21.

Near-wall refinement has been applied only for the hexahedral fine grid without reaching *y+* values compatible for explicit near-wall solution, so wall functions have been adopted for all cases. This approach is anyway

justified when dealing with realistic ICE flows, where the computational cost for the explicit resolution of the near-wall region would be prohibitive. The y^+ values has been anyway contained within a range between 50 - 80, depending on the grid.

Time discretization has been performed in light of the considerations made in Chapter 2 and a timestep in the range of 10^{-6} - 10^{-7} s has been adopted, ensuring a maximum $Co < 0.7$, as a compromise between accuracy and computational cost.

Inlet boundary conditions have been set imposing a constant axial velocity profile of 65 m/s matching the experimental bulk flow velocity. Here, as done for the Dellenback case, no turbulent fluctuations have been explicitly set. In this case, the “pseudo-laminar” inlet boundary condition has been justified by the length of the inflow duct which allows for a natural onset of turbulence well before reaching the valve. At the flow bench outlet a homogeneous zero-gradient velocity boundary condition and a fixed value of zero relative pressure have been set. No-slip conditions have been applied at all walls.

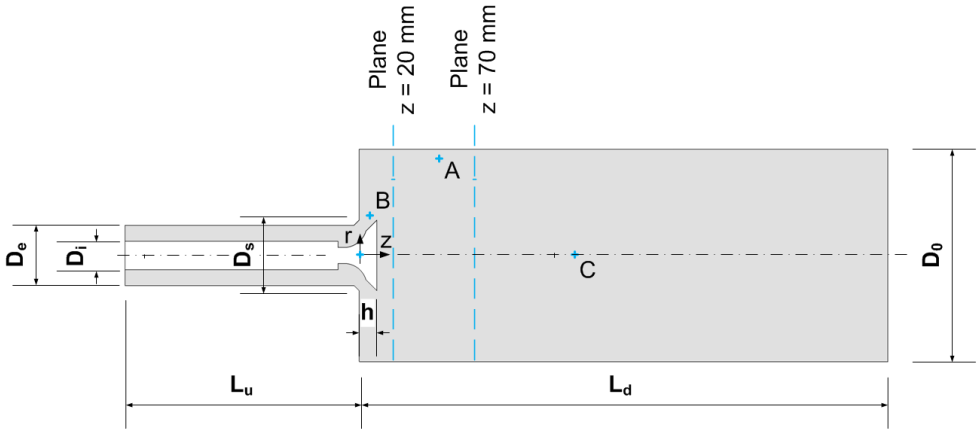


Figure 4.20: Geometry of the Thobois test case and probe locations.

Parameter	Symbol	Value
Upstream length	L_u	130
Downstream length	L_d	300
External diameter	D_e	34
Internal diameter	D_i	17
Main diameter	D_0	60
Valve seat diameter	D_s	40
Valve lift	h	10

Table 4.1: Main geometric parameters of the Thobois test case.

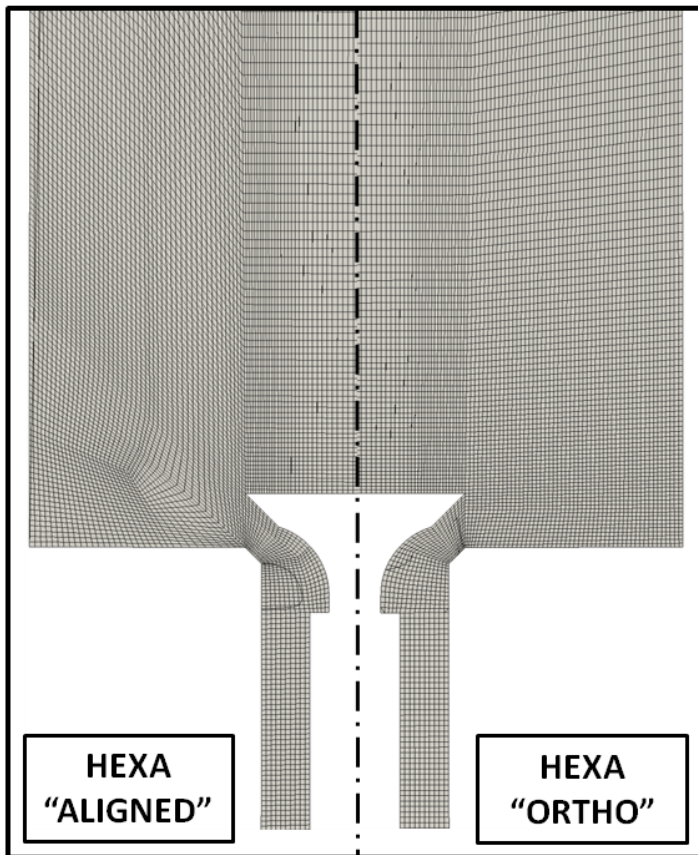


Figure 4.21: Coarse hexahedral grids for the Thobois test case.

4.2.2 Results - Velocity profiles

The evaluation of the analyses results has been made by comparing the velocity mean and RMS radial profiles at two different locations with the corresponding experimental data. The two sampling stations are placed respectively at 20 *mm* and 70 *mm* from the expansion. The computational profiles have been averaged over a suitable amount of mean flow advection times, sufficient to guarantee statistical convergence, and circumferentially averaged. To be noted that the experimental profiles have been mirrored across the duct centerline. A certain degree of asymmetry appears in such results whose origin could be attributed to measurement errors or flow asymmetry due to geometric misalignment in the duct-valve assembly or in the inflow conditions.

Figure 4.22 displays the velocity profiles and the corresponding turbulent fluctuation levels obtained with the different SGS models on the coarse grid. The evaluation of the profiles reveals an overall good agreement with the measurements with some deficiencies of the fluctuations RMS close to the walls. The tangential profiles in particular are almost always under-predicted. All the SGS models exhibit similar behaviors but overall better performances are reached by the WALE and the Sigma models, while the dynamic Smagorinsky seems to be the less accurate one.

Figure 4.23 shows the velocity profiles obtained with the Sigma SGS models for the two flow-aligned hexahedral grids. Both the coarse and the fine density grids seem to accurately resolve the average velocity profiles. While the RMS profiles of the fine grid present a better agreement for the 20 *mm* station, the accuracy level is almost identical for the 70 *mm* station. Interestingly the tangential component of the fluctuations continues to be underestimated on the 70 *mm* station even when employing the finer mesh.

The effect of the element type on the solution is evaluated through Figure 4.24 which displays the velocity profiles obtained with the Sigma SGS model on the coarse flow-aligned hexahedral grid and on a uniform tetrahedral grid with almost the same number of cells. Both types of cells allow for a proper resolution of the mean velocity profiles but the tetrahedral grid performs poorly in the evaluation of the fluctuations. In particular, the RMS profiles of the fluctuations of both the axial and the tangential components of the velocity fail to resolve the peaks present at $r/R0 = 0.5$ at the 20 *mm* station. The behavior of the tetrahedral grid is here somewhat different from the one presented in the Dellenback case where the stochastic velocity components were predicted reasonably well. A possible explanation for this deficiency is the greater level of numerical diffusion arising

with the use of tetrahedral elements for the resolution of strongly oriented flows such as the jet across the valve. When the jet has expanded and the velocity field gradients are smaller, then, the tetrahedral grid reverts back to resolution capabilities comparable to the hexahedral one, as suggested by the RMS profiles at the 70 *mm* station.

Finally, Figure 4.25 presents the comparison of the results obtained on the two hexahedral coarse grids, namely the “aligned” and “ortho” grids. As aforementioned, such grids differ only for cells orientation respect to the jet flow. The profiles show clearly that there are no significant differences in the velocity field resolution obtained by the two discretization approaches. This result is quite interesting considering that in realistic ICE configurations it is not always practically feasible to align the grid with the valves jets. For instance, some ICE CFD packages use octree-type hexahedral unstructured grid generators which cannot ensure such flow alignment.

4.2.3 Results - Quality assessment

The Thobois LES analyses quality assessment has been performed through the parameters proposed in Chapter 3, following the methodologies described for the Dellenback test case.

The M parameter comparison is visualized in Figure 4.26. The picture reports the maps of the parameters for the two flow-aligned hexahedral grids. Apparently the increment in resolution level achieved by using the finer grid is marginal. This is consistent with the velocity profiles comparison reported in the previous section, where only minor differences were noticeable between the two sets. The M maps show anyway slight resolution improvements in the inflow duct and in the stagnation region past the valve. It must be noted also that, even with the coarse mesh, the M parameter is always smaller than the 20% threshold in basically the whole domain past the expansion and far from walls.

Figure 4.27 shows the LSR parameter contour for the aforementioned grids. The scale resolution improves with grid refinement and the LSR values for the bulk regions past the expansion are always smaller than 3 with some exception for the coarse grid in the jet region. Near the walls both grids suffer a lack of resolution and this was somehow expected due to the wall modeling approach adopted in the simulations.

Viscosity ratio contours are reported in Figure 4.28 where the contribution of the sub-grid scales lowers with grid refinement. High levels of turbulent viscosity tend to remain in the jet region and close to walls for the lack of resolution that the near wall modeling determines .

4.2.4 Results - Energy spectra

Energy spectra of the resolved flow field have been evaluated at different locations in the domain (resp. points A, B and C) as shown in Figure 4.20.

Figure 4.29 shows the effect of the grid density on the resolved kinetic energy spectrum at point A. Both the coarse and fine hexahedral grids exhibit a $-5/3$ power law inertial subrange behavior but the denser grid seems to follow better the asymptotic trend and to present a slightly higher cutoff frequency, as expected.

Figure 4.30 visualizes the influence of the element shape on the energy spectrum evaluated at point A. Even if both grids follow the asymptotic power law behavior, the tetrahedral one seems to be noisier and more damped than the hexahedral one.

The SGS model effect on the spectrum shape for point A is reported in Figure 4.31, where a substantial lack of influence of the used model emerges.

Finally, Figure 4.32 shows the energy spectrum shapes at points A, B and C. While points A and C appear to be sufficiently well resolved, point B exhibits an unusual behavior towards the end of the spectrum. Such ambiguous rise in the energy level has to be probably ascribed to the lack of resolution that characterizes this region, as mentioned in Paragraph 4.2.3.

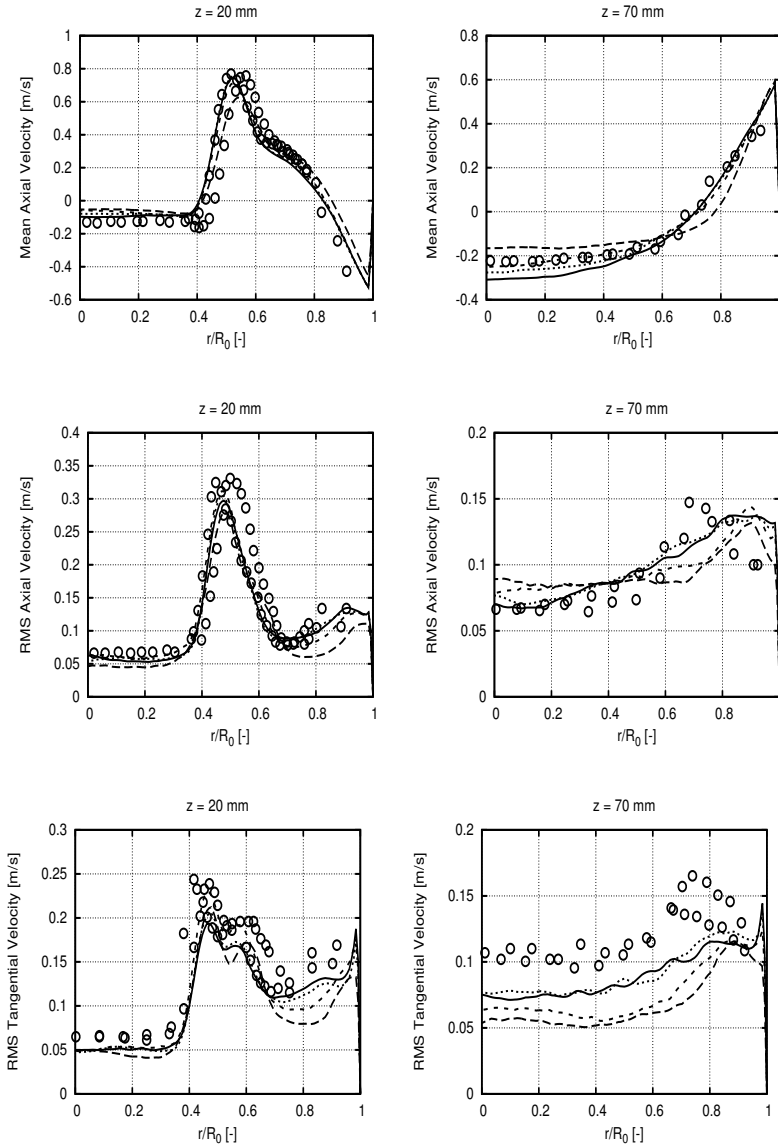


Figure 4.22: Mean and RMS velocity profiles for different grid densities (Sigma model - coarse grid). Legend: \circ experiment, $---$ dynamic Smagorinsky, $- - -$ dynamic WALE, $—$ WALE, \cdots Sigma.

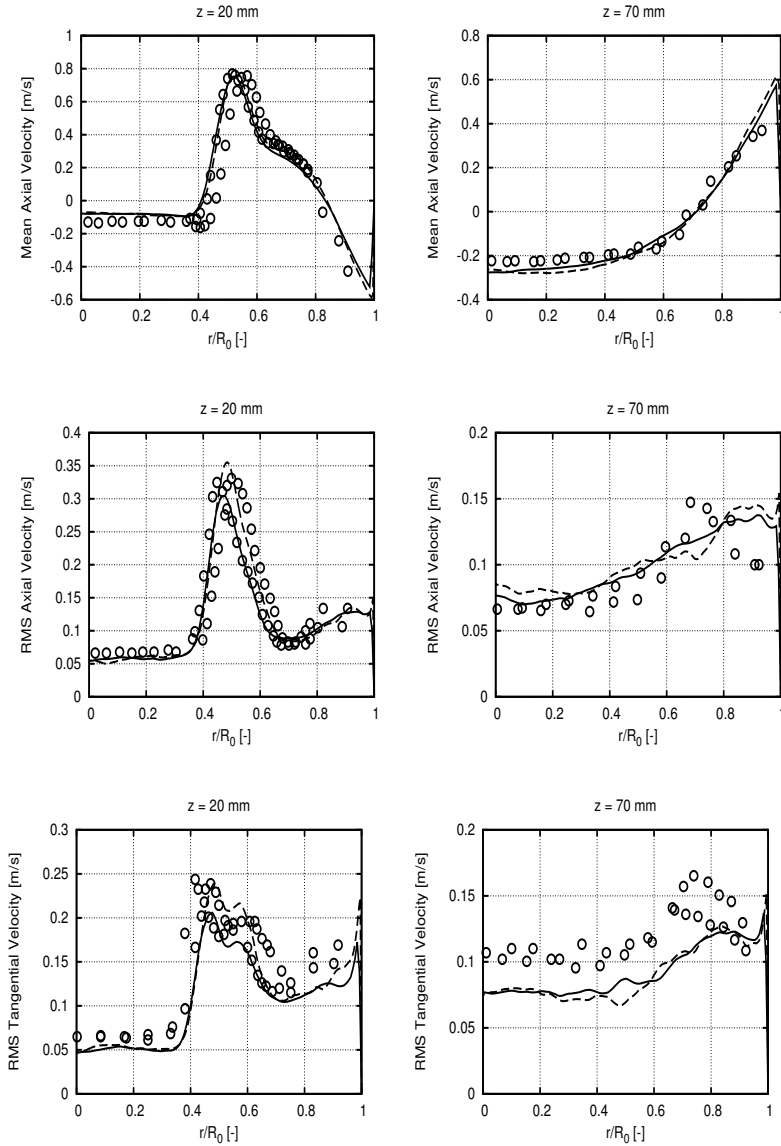


Figure 4.23: Mean and RMS velocity profiles for different grid densities (Sigma model - coarse grid). Legend: \bigcirc experiment, — coarse grid, - - - fine grid.

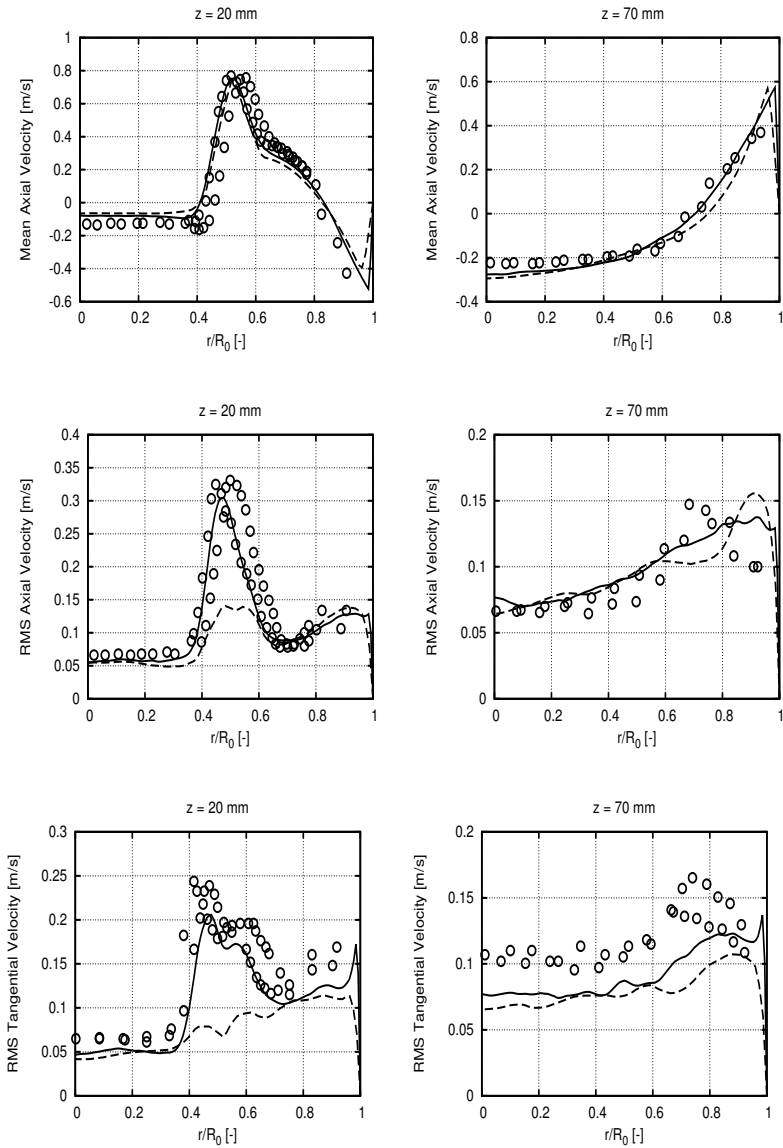


Figure 4.24: Mean and RMS velocity profiles for different grid element types (Sigma model - coarse grid). Legend: \circ experiment, — hexa, - - - tetra.

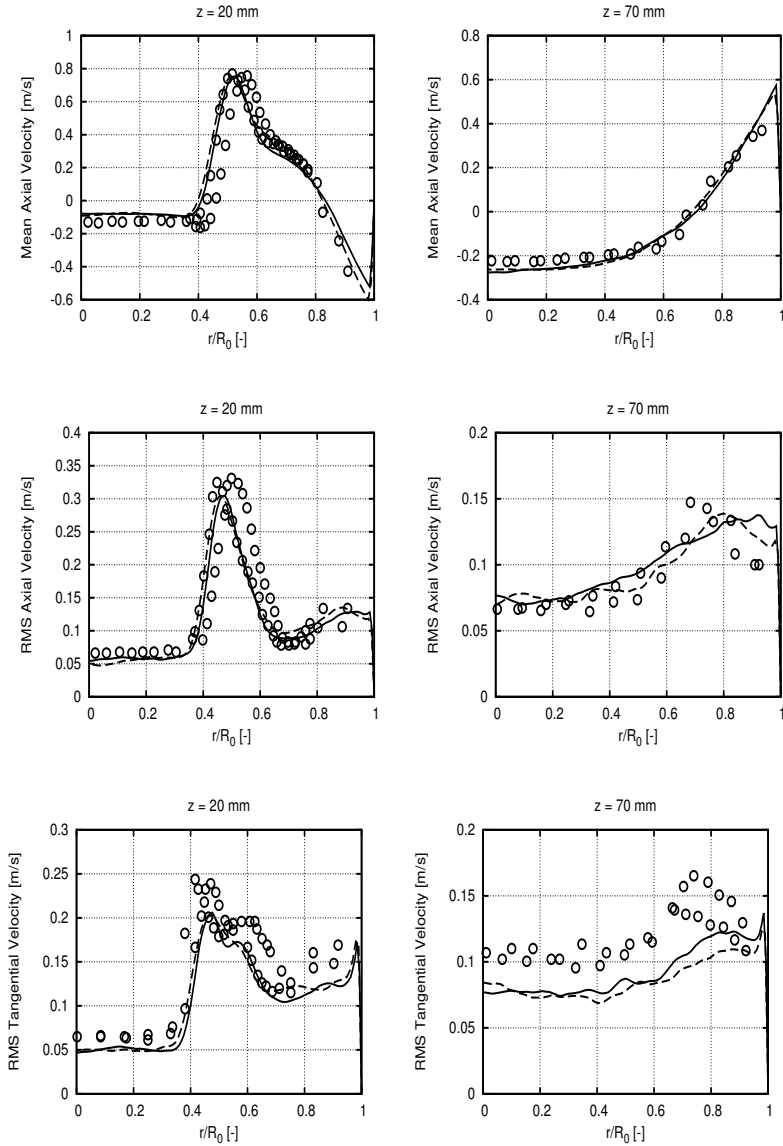


Figure 4.25: Mean and RMS velocity profiles for different grid alignments (Sigma model - coarse grid). Legend: \circ experiment, — coarse orthogonal, - - - coarse aligned.

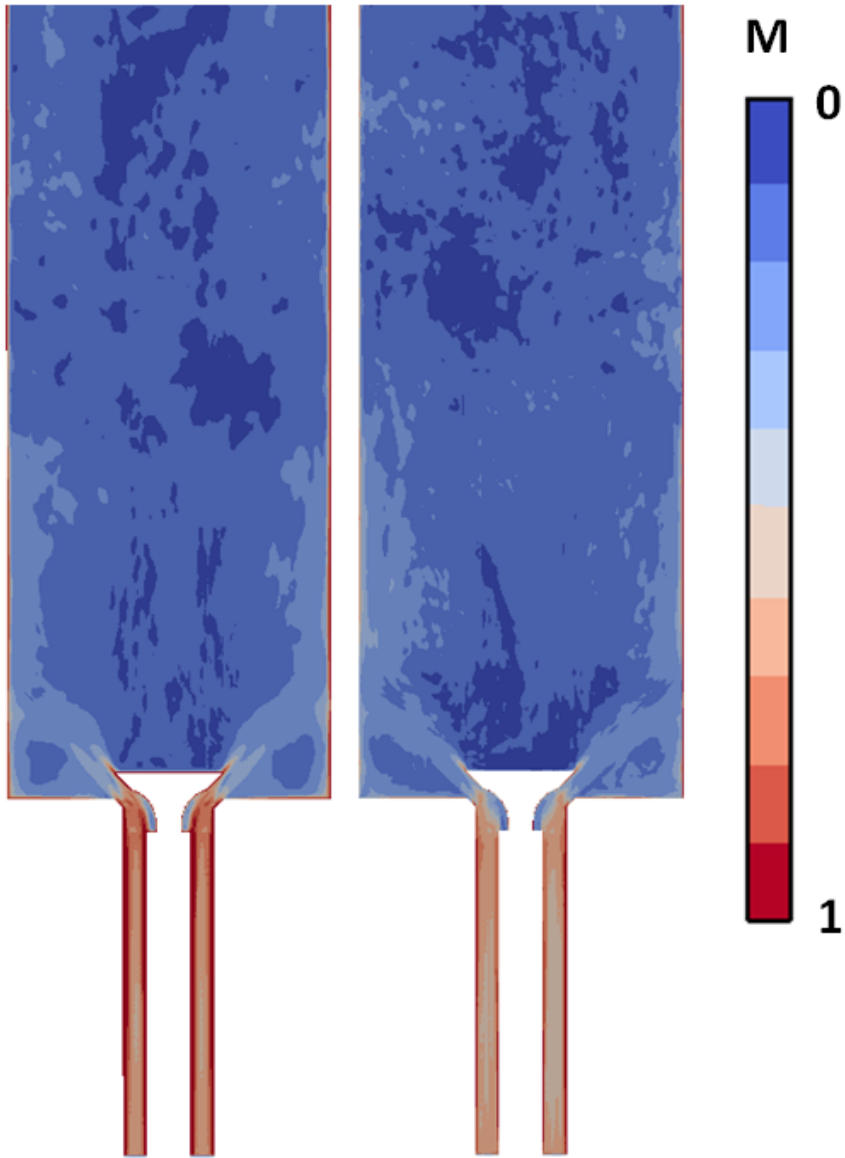


Figure 4.26: M parameter for different grid densities (Sigma SGS model): left - coarse grid, right - fine grid.

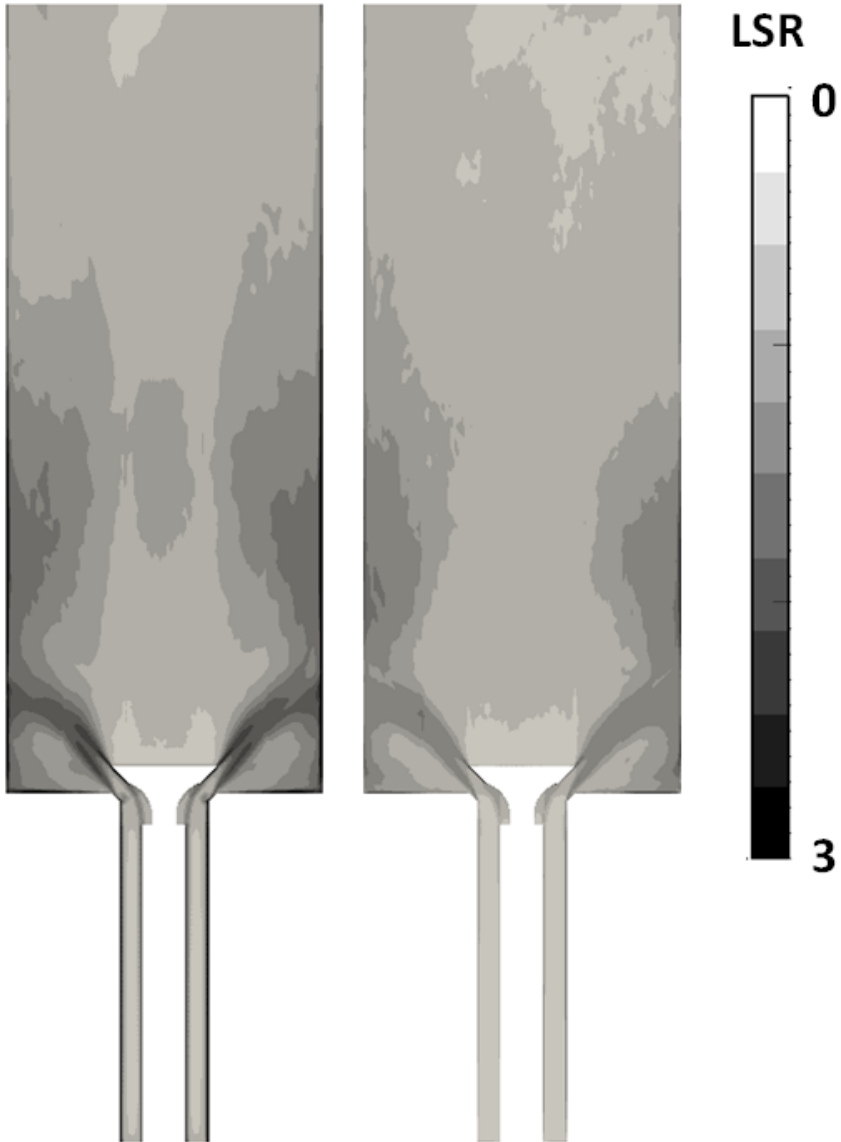


Figure 4.27: *LSR parameter for different grid densities (Sigma SGS model): left - coarse grid, right - fine grid.*

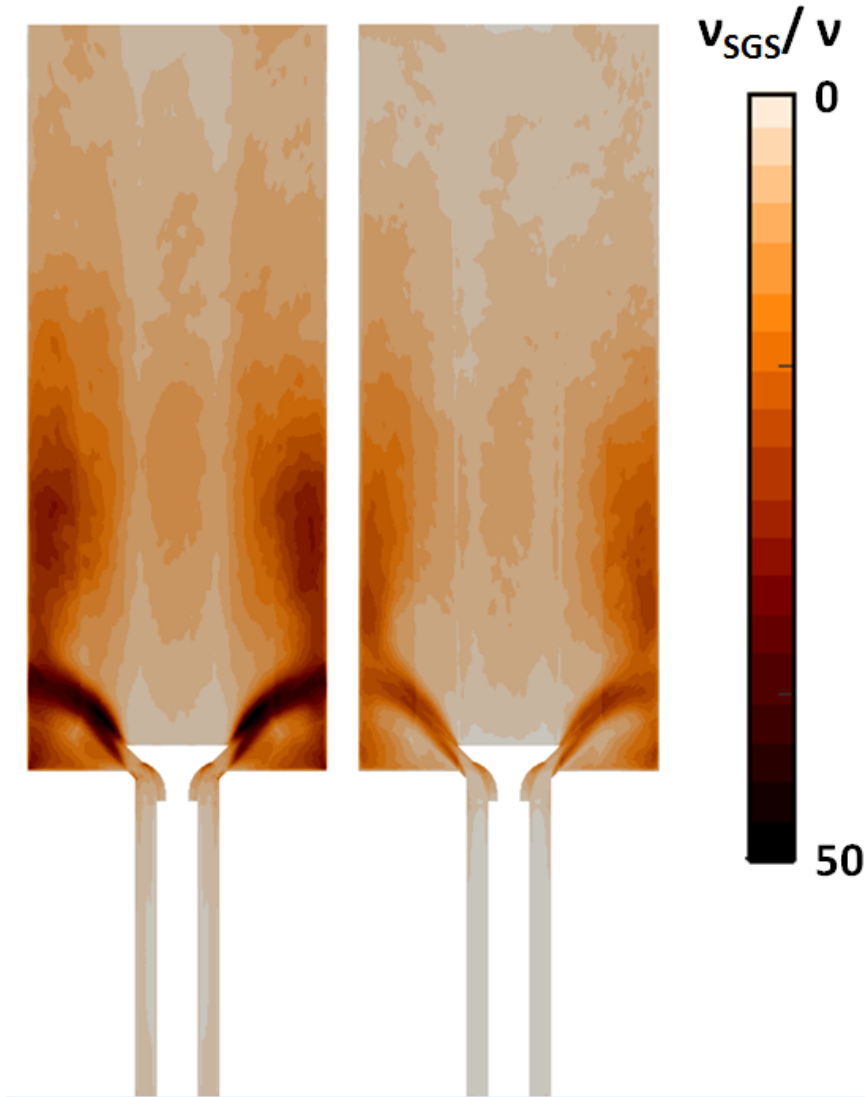


Figure 4.28: ν_{SGS}/ν parameter for different grid densities (Sigma SGS model): left - coarse grid, right - fine grid.

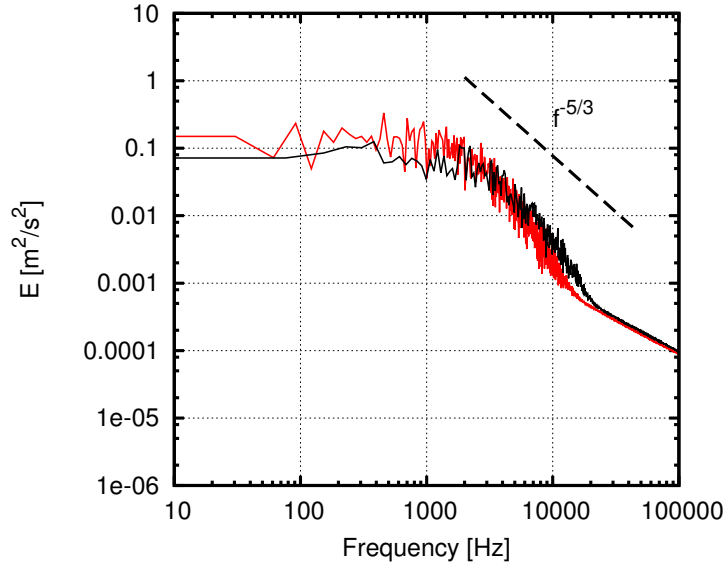


Figure 4.29: Turbulent kinetic energy spectra for different grid densities (Sigma SGS model): coarse grid (red), fine grid (black).

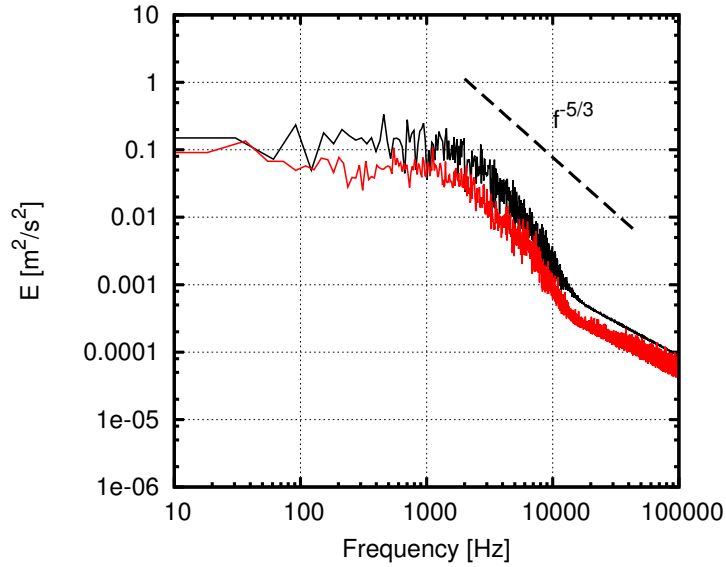


Figure 4.30: Turbulent kinetic energy spectra for different grid element type (Sigma SGS model): tetrahedral grid (red), hexahedral grid (black).

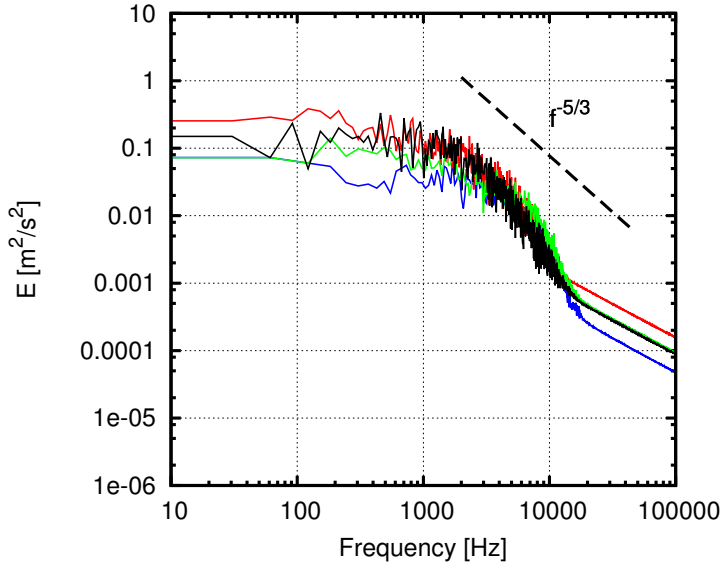


Figure 4.31: Turbulent kinetic energy spectra for different SGS models: dynamic Smagorinsky (blue), WALE (red), dynamic WALE (green), Sigma (black).

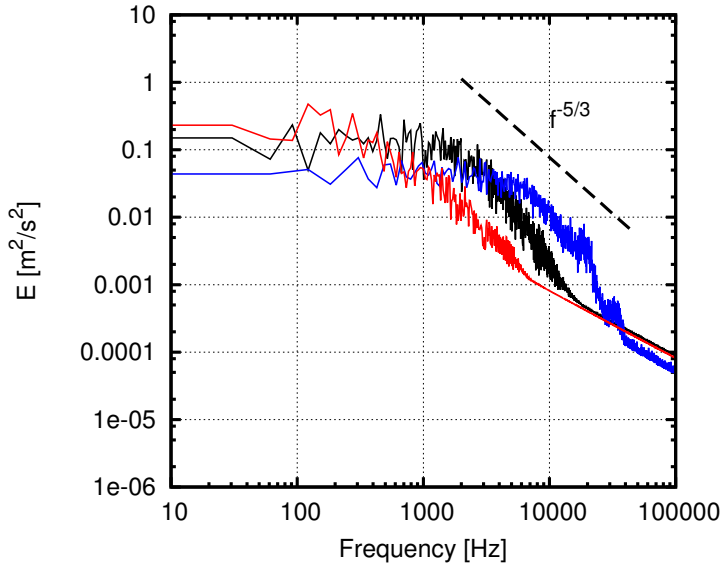


Figure 4.32: Turbulent kinetic energy spectra for different locations: point A (black), point B (blue), point C (red).

4.3 Computational aspects

The analyses presented in this chapter have been run on the following hardware:

- 2 x Intel Xeon E5-2609 @ 2.40GHz workstation.
- 4 x AMD Opteron 6212 @ 2.60GHz blade.

OpenFOAM® v 2.3.0 has been compiled with the GNU GCC v. 4.8.1 linux compiler suite and with Open-MPI v. 1.7.2.

A rough estimate of the relative computational cost of the different SGS models is reported in Table 4.2. The tests have been performed on the Intel workstation employing the medium density grid for the Dellenback case with a serial run. The differential in computational cost of employing the various SGS models has been evaluated counting the number of iterations (time-steps) performed for the same fixed wall clock time. Such estimate is reported in the table as "Computational Intensity". While running the tests, the memory requirement of each run has been tracked. Both the Computational Intensity and the memory footprint (here "Memory Requirement") have been normalized accordingly to the dynamic Smagorinsky SGS model performance. The WALE model seems to be the less demanding SGS model, while the Sigma appears to be the most expensive. For this latter model, it is in the author opinion that the computation of the velocity derivatives tensor eigenvalues is the main cause for the high computational demand. It is however important to note that these tests have been carried out without claiming to be rigorous assessments but just to provide a rough idea of how much does it cost to use a certain model respect to another.

Sub-Grid Scale Model	Dynamic Smagorinsky	WALE	Dynamic WALE	Sigma
Memory Requirement	1	1.01	1.05	1.05
Computational Intensity	1	0.77	0.85	1.38

Table 4.2: Computational requirements of the different SGS models.

The parallel performance of the code have also been tested and Figure 4.33 shows the scalability of *OpenFOAM*® on the Opteron blade in use. As it can be seen the code speedup is almost linear with the number of parallel processes employed for the run. To be noted also that the deviation from the ideal trend occurring at the last computational point has to be probably

attributed to the pretty small amount of cells (< 50000) for each partition, thus enhancing inter-process communication time.

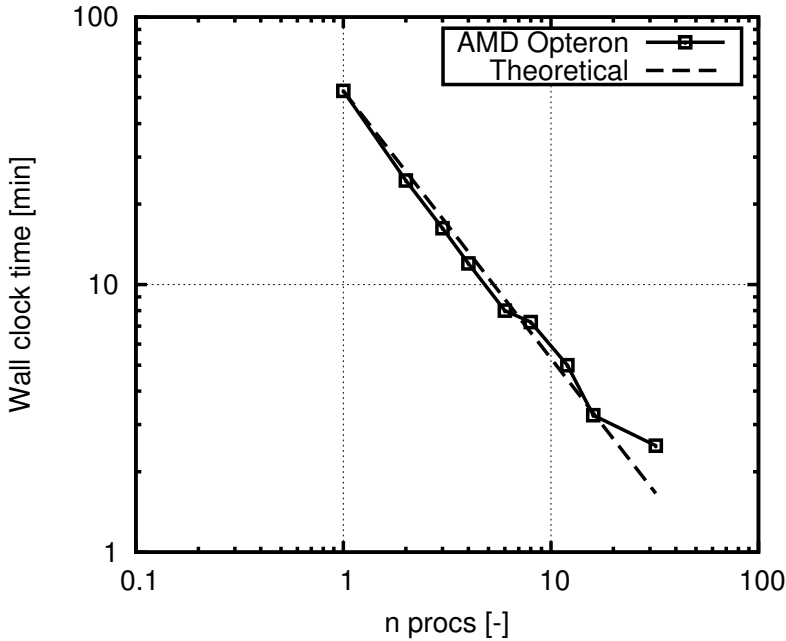


Figure 4.33: *OpenFOAM[®] scalability tests performed on the quad-proc. AMD Opteron 6212 blade.*

4.4 Closure

The present chapter has presented the assessment of the SGS turbulence models implemented into the open-source CFD code *OpenFOAM[®]* v. 2.3.0, extending its basic capabilities. The tested SGS models are the dynamic WALE and the Sigma SGS models, whose details have been presented in Section 2.4 and are here proposed as advanced/improved alternatives to the WALE or the dynamic Smagorinsky models, being these latter the most common SGS modeling choices nowadays available in commercial CFD codes. The solver predictive capabilities have been then assessed performing LES analyses of the Dellenback abrupt expansion and of the Thobois flow bench test cases.

The abrupt expansion computational results have been compared to the experimental data available and an overall good agreement between the two data sets has been found. The Sigma model performed better than

the dynamic WALE and its predictive capabilities are comparable to the WALE model. Both the implemented SGS models outperform the dynamic Smagorinsky model which demonstrated poor accuracy in wall regions. Discrepancies between experiments and simulations appear when the models are run on fine grids, in particular at the pipe axis region. This behavior has to be probably attributed to an improper tuning of the models constants but this hypothesis needs further investigation in order to be confirmed.

The flow bench simulations exhibited an overall good matching with the experimental results. Here again, the Sigma model performed better than the dynamic WALE and its predictive capabilities are basically the same of the WALE model. Poor prediction of the fluctuating velocity components are obtained with tetrahedral elements, probably due to excessive numerical diffusion in the valve jet region. Interestingly, when different grid orientations respect to the valve jet have been compared, they proved to provide similar resolution capabilities with no degradation in solution accuracy .

LES quality has been evaluated through a series of one-grid and two-grid estimators. The main finding is that such estimators present similar distributions for all grids leading to qualitatively comparable indications for all the SGS models tested. The only differences emerge in terms of range of values, sensitivity and easiness of applicability. The finer grids proved to provide sufficient resolution with respect to the purpose of the current investigation with some deficiencies at walls where wall functions have been used.

The computational costs and the memory requirements related to the adoption of the different SGS models have been also investigated. The Sigma model resulted the more demanding model while the dynamic WALE requirements lie in between the ones of the two reference models, i.e. the dynamic Smagorinsky and the WALE models.

In light of the findings presented in this chapter, it can be concluded that the Sigma SGS model looks promising for applications to LES of ICE flows. However, it needs to be further optimized in order to reduce its computational costs, in particular when compared to the standard WALE model. Conversely, the dynamic WALE model hasn't performed as well as expected and it requires further development in order to overcome some predictive deficiencies.

CHAPTER 5

Virtual Flow Bench

5.1 Flow bench general notes

Engine coherent flow structures such as swirl and tumble motions are key factors for the combustion process due to their capability to rise turbulence levels and enhance mixing which, in turns, severely influence both fuel efficiency and pollutant emissions. Automotive industry has therefore put great efforts over the last decades in evaluating air flow during induction stroke and air flow within the cylinder.

Nowadays permeability, swirl and tumble motion characterizing a specific cylinder head are evaluated experimentally at design stage mainly using stationary flow benches. Such tests allow characterizing each head prototype using non-dimensional parameters like flow coefficient, swirl and tumble ratios and, finally, to compare and optimize the different designs.

The layout of a typical induced swirl flow test bench is illustrated in Figure 5.1. The intake system object of study is mounted on a dummy cylinder which terminates in a settling plenum. A blower connected to such plenum induces the air inflow from the engine head. The air proceeds through the cylinder and expands in the plenum and is finally discharged back in the ambient through the blower duct. For each valve lift of interest the system

is run until overall steady flow conditions are reached. Mass flow rate is then acquired, usually through an orifice flow rate meter, while the swirl torque is measured using a swirl meter which can be either a paddle wheel or an impulse swirl meter.

Even if no unified testing methodology exists [70], in the common practice a number of measurement is required for characterizing a single head. The tests are repeated varying the valve lift by equal fractions of the maximum valve lift for a total of 10-12 lifts. The data acquired in terms of flow coefficients and swirl/tumble torque is then properly processed accordingly to specific procedures in order to determine the swirl/tumble ratio, which synthetically describes the tested head performance (see [46], [71] or [72] for reference) .

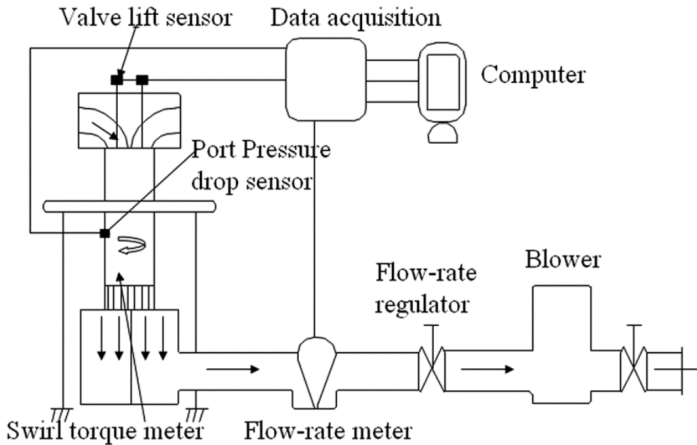


Figure 5.1: *Generic induced steady flow bench scheme. Reproduced from Lu [73].*

The prototype head object of the present study belongs to a 4 valves per cylinder heavy duty Diesel engine manufactured by *VM Motori*, who kindly made also available the experimental results. Geometric details of the simulated engine are summarized in Table 5.1.

This engine head has been already tested in a previous study [74] where four different prototype engine heads have been simulated with a commercial code of reference (*AVL FIRE*[®]) adopting a standard RANS modeling. The particular head chosen for the present study has been the one presenting the poorest agreement in terms of predicted swirl torque. Here, it is firstly simulated using *OpenFOAM*[®] and a standard RANS approach to compare the predictive capabilities of the open-source code against the commercial one. Then, the LES methodology is applied in order to better understand the system fluid dynamics and to investigate the causes of potential RANS

predictive deficiencies. Finally, the insights obtained through LES are employed to propose a different RANS approach overcoming or, at least, alleviating the aforementioned predictive deficit.

Parameter	Units	Value
Engine type	[-]	Diesel
Strokes	[-]	4
N° valves	[-]	4
Bore	[mm]	104
Stroke	[mm]	115.4
Intake valve diameter	[mm]	35
Intake valve seat diameter	[mm]	27.3
Intake valve lift (max)	[mm]	10

Table 5.1: *Engine parameters.*

5.2 Eddy-viscosity RANS approach

5.2.1 Case setup

The computational domain is reported in Figure 5.2 and it comprises the intake ducts, the engine head, the intake valves and the dummy cylinder liner. The domain discretization has been performed using *cfMesh*[®] [75], an open-source meshing program for *OpenFOAM*[®]. For each valve lift a different geometry has been used. The hexa-dominant computational grids so obtained reach approximately 2.1 M cells and they have been locally refined at walls in order to properly handle boundary layers for which two wall layers have been employed. Grid details are shown in Figure 5.3. The mesh base size is 1.8 mm and several local refinements have been applied to the computational grid in order to properly resolve the flow features. In particular one fourth of the base mesh size has been used close to the walls and up to one eighth for the valve throat walls.

The solver employed is *OpenFOAM*[®] v 2.3.0. and a steady compressible formulation has been adopted. The fluid used has been air at ambient conditions, modeled as ideal gas. For the convection and diffusion terms a second order upwinded scheme has been employed. The pressure-velocity coupling has been performed using the *SIMPLEC* algorithm ([8], [10]). A standard eddy-viscosity RANS approach has been here used employing a high-Re *RNG* – $k\varepsilon$ model [76] for turbulence modeling and wall functions have been used to treat fluid flow at solid surfaces.

The experiments have been run adopting a fixed pressure drop of 500 mm_{H₂O}, therefore the same pressure drop is applied to the CFD model.

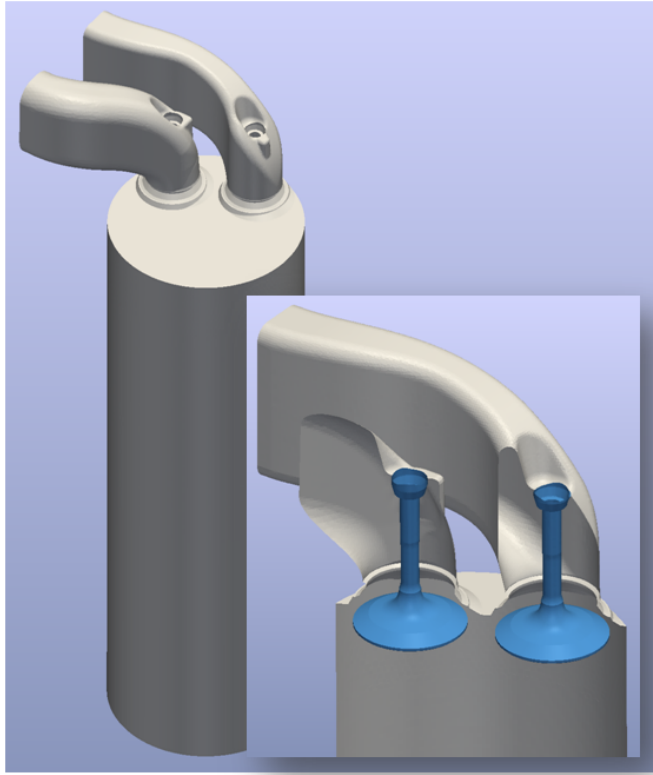


Figure 5.2: *Computational domain geometry.*

On the ducts inlets a fixed total pressure condition is prescribed, while at the dummy cylinder outlet a static pressure value is set along with a zero-gradient velocity condition. The velocity inflow boundary condition on the ducts is obtained from the patch-face normal component of the internal-cell value. For the turbulent variables k and ε a zero-gradient outflow is specified at the cylinder outlet while, on the ducts inlets, turbulence characteristics have been set through the duct hydraulic diameter and prescribing a turbulent intensity of 5%. Table 5.2 summarizes the boundary conditions applied for the RANS simulations.

The domain is initialized setting a uniform static pressure field close to the inlets static pressure value and the outlet pressure is progressively ramped-down to its target value in order to smoothly develop the fluid flow. This steering procedure is often necessary to stabilize stationary compressible simulations presenting non-negligible pressure differences in the domain.

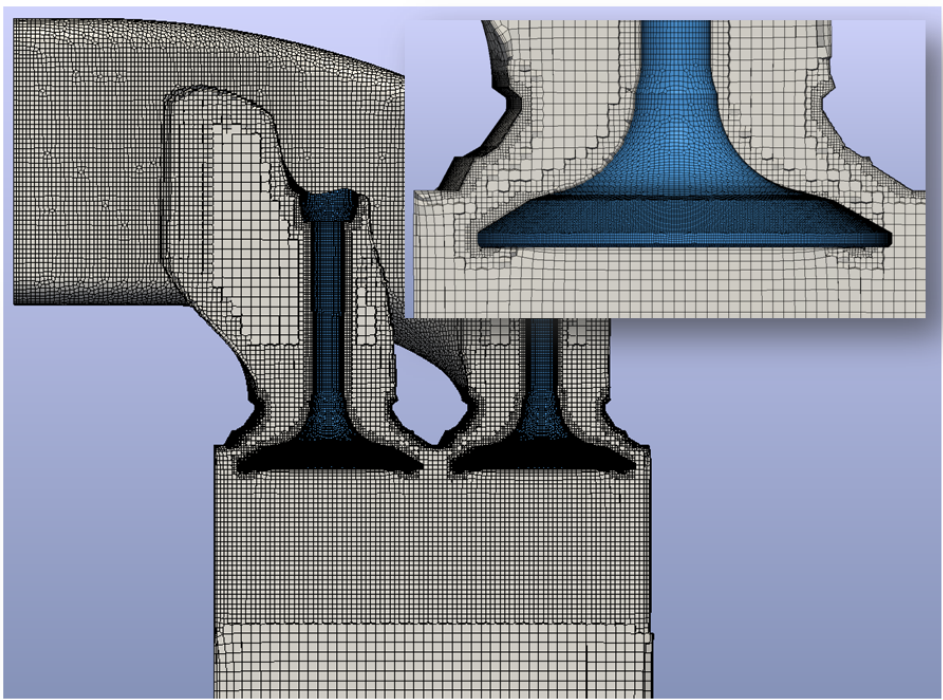


Figure 5.3: *Flow bench computational grid.*

Boundary	Condition type	Variable	Condition
Inlets	Total pressure inflow	U	Pressure driven inflow
		p	Total pressure = 100800 Pa
		T	Static temperature = 298.15 K
		k	Turbulence intensity = 5%
		ε	Turbulence mixing length = 0.5 mm
Outlet	Static pressure outflow	U	Outflow
		p	Static pressure = 95897 Pa
		T	Outflow
		k	Zero gradient
		ε	Zero gradient
Walls	No-sip wall	U	Wall function managed
		p	Zero gradient
		T	Zero gradient (Adiabatic)
		k	Wall function managed
		ε	Wall function managed

Table 5.2: *Virtual flow bench boundary conditions (RANS).*

5.2.2 Impulse swirl meter modeling

VMMotori carried out the experimental campaign on the head prototypes employing an impulse swirl meter, accordingly to Ricardo methodology [71]. The instrument, sketched in Figure 5.4, from a fluid-dynamic perspective, acts damping out tangential velocity components and causing at the same time a minimal pressure drop across itself. In order to reproduce the presence of the impulse swirl meter, a porous medium has been introduced inside the cylinder as it can be seen in Figure 5.5. The porous medium has been modeled using the Darcy-Forchaimmer model available in *OpenFOAM*® [38]:

$$\frac{dp}{dz} = -\alpha\mu w - \beta\frac{\rho}{2}w^2 \quad (5.1)$$

where z is the cylinder axis direction, w is the velocity component along such direction and the porosity parameters α and β have been determined experimentally by *VMMotori* and kindly provided.

Accordingly to Ricardo testing methodology, the impulse swirl meter has to be placed at a distance of $1.75 \times \text{Bore}$ from the head. Therefore the porous region reproducing it begins at $1.75 \times \text{Bore}$ from the head extending for 50 mm inside the cylinder. Finally other 100 mm are left past the porous medium to stabilize outflow conditions.

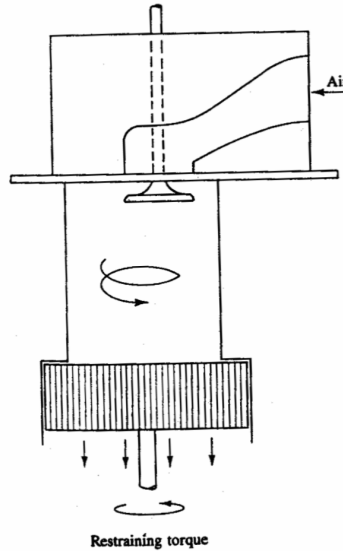


Figure 5.4: Schematic of steady-flow impulse swirl meter. Reproduced from Heywood [46].

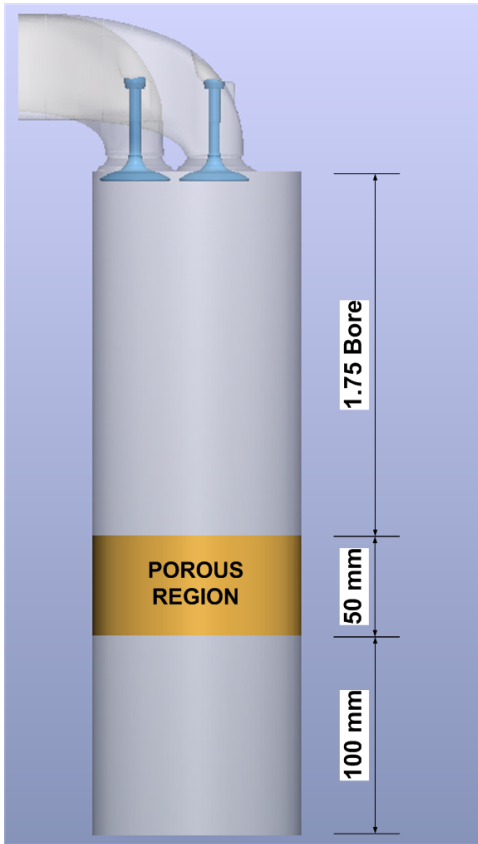


Figure 5.5: *Porous region location.*

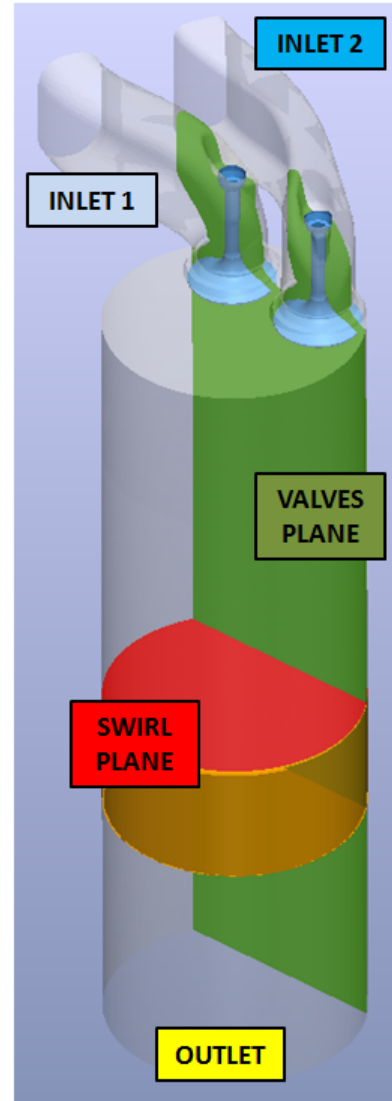


Figure 5.6: *Post-processing topology used for the Virtual Flow Bench.*

5.2.3 Results

Post-processing of results dealt primarily with the evaluation of mass flow rate and torque. Both quantities have been evaluated through a cut plane perpendicular to cylinder axis and located 5 mm upstream the porous region to avoid possible disturbances in the sampling. Since some cases showed a mass flow rate slightly oscillating about a mean value, the average over the last 8000 iterations have been taken. The torque has been evaluated through the procedure described in Appendix B.

System mass flow rate and swirl torque for each valve lift are synthesized in Figures 5.8 and 5.9 respectively, while numerical values are reported in Tables 5.4 and 5.5. Mass flow rates exhibit a pretty fine agreement with the experimental data, reporting a maximum relative error of 10% at lift 1 mm. Conversely, the swirl torque prediction is quite poor and the mismatch with experimental data tends to growth with the valve lift. The maximum relative error is about 30% in this case. The torque trend is roughly similar to the experimental one, increasing with the lift, even if not monotonically.

Figures 5.8 and 5.9 include also the results obtained with *AVL Fire*[®] and it is evident how the two codes perform in a quite similar fashion in terms of predictive capabilities. They both resolve reasonably well the mass flow rate but fail in the torque estimation, at least at medium/high lifts.

This kind of results is not surprising since several scientific publications present similar findings: a reasonable accuracy is reached in the head permeability evaluation but there is pretty low reliability in swirl torque evaluation and mismatches up to 50% in the prediction of the latter can be found as reported in [77] or [78].

The effect of the swirl meter (modeled as a porous region) on the flow is visualized in Figure 5.10 where it is evident the straightening action that it exerts on the streamlines. The swirl meter virtually splits the fluid domain into two main regions. Upstream the porous region the flow field exhibits noticeable vortical structures some of which are aligned in a helical fashion with the cylinder axis, while others are normal to the axis and are presumably generated by the interaction between the valve jets and the cylinder walls. Downstream the porous region the streamlines appear straight and aligned with the cylinder axis with no distinct vortex visible.

Figure 5.11 presents a map of the velocity magnitude field on the valves plane (see Figure 5.6 for post-processing topology). Most of the fluid domain presents relatively low velocities, of the order of 10÷20 m/s, while higher values can be found only in proximity of the valves, as expected.

The port fluid jets can reach velocities up to 100 m/s and generate a series of vortices as discussed in the previous paragraph. Behind the valves local recirculating regions are present. The two cutplanes presented in the same Figure show streamlines of the velocity field projection along the axis-normal direction at two different locations through the cylinder. The top slice, made at $0.8 \times \text{Bore}$ from the head surface, presents two distinct counter-rotating vortical structures. Downstream, in the bottom slice, the flow field has developed and a single clockwise rotating vortex is left.

Figure 5.7 shows flow detachments of the jet occurring at the ports and wide recirculating regions past the valves. Minor flow recirculations are also highlighted in the valve ports.

Flow field details in the ports region are visualized in Figure 5.12. Several features of the flow already discussed in the previous paragraphs are present. Moreover, the map reveals an extensive jet attachment to the liner walls close to the head with pretty strong associated velocity gradients. This, in turns, leads locally to high values of turbulent kinetic energy as shown in Figure 5.13.

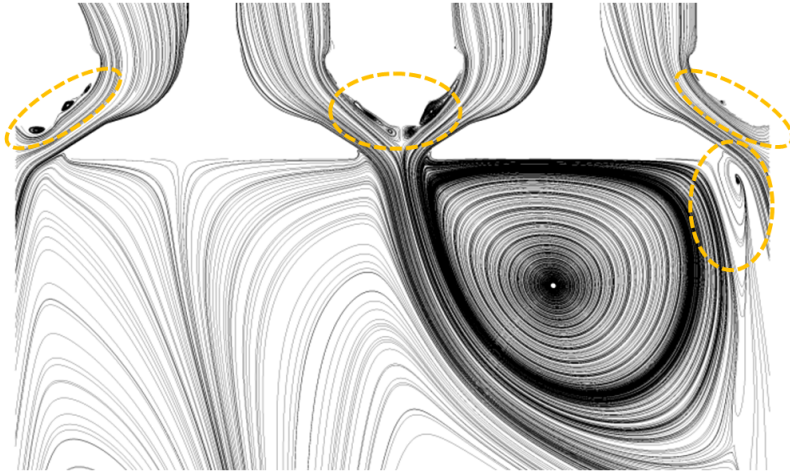


Figure 5.7: Details of the flow in the valve region. Local fluid recirculations and flow detachments highlighted.

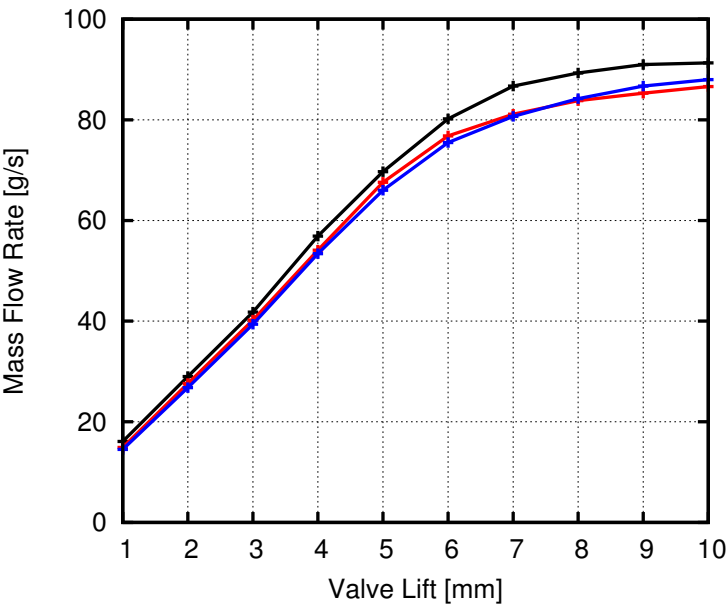


Figure 5.8: Mass flow rate as a function of valve lift: experiments (black), AVL Fire[®] (red), OpenFOAM[®] (blue).

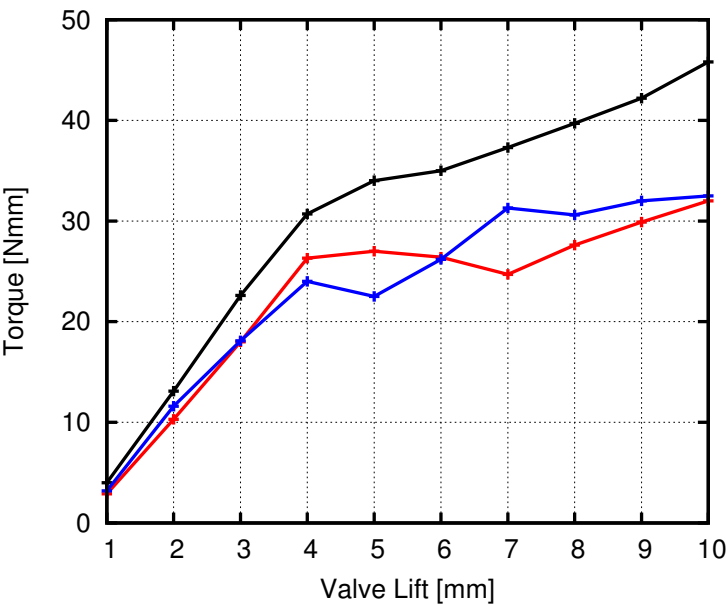


Figure 5.9: Swirl torque as a function of valve lift: experiments (black), AVL Fire[®] (red), OpenFOAM[®] (blue).

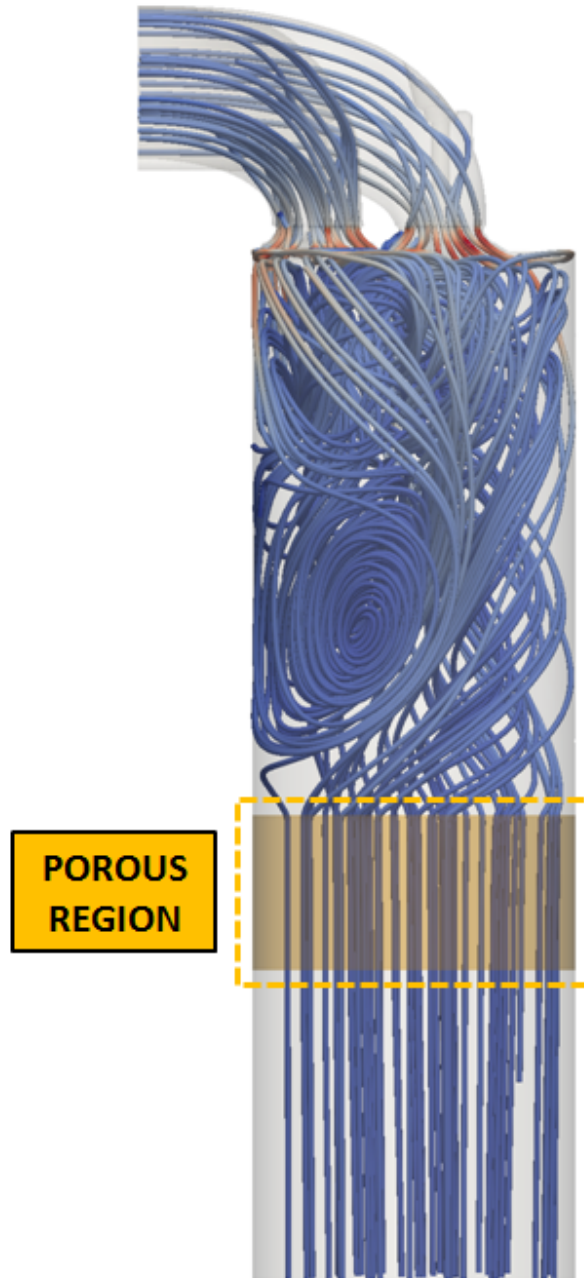


Figure 5.10: *Visualization of the porous region flow straightening through streamlines.*

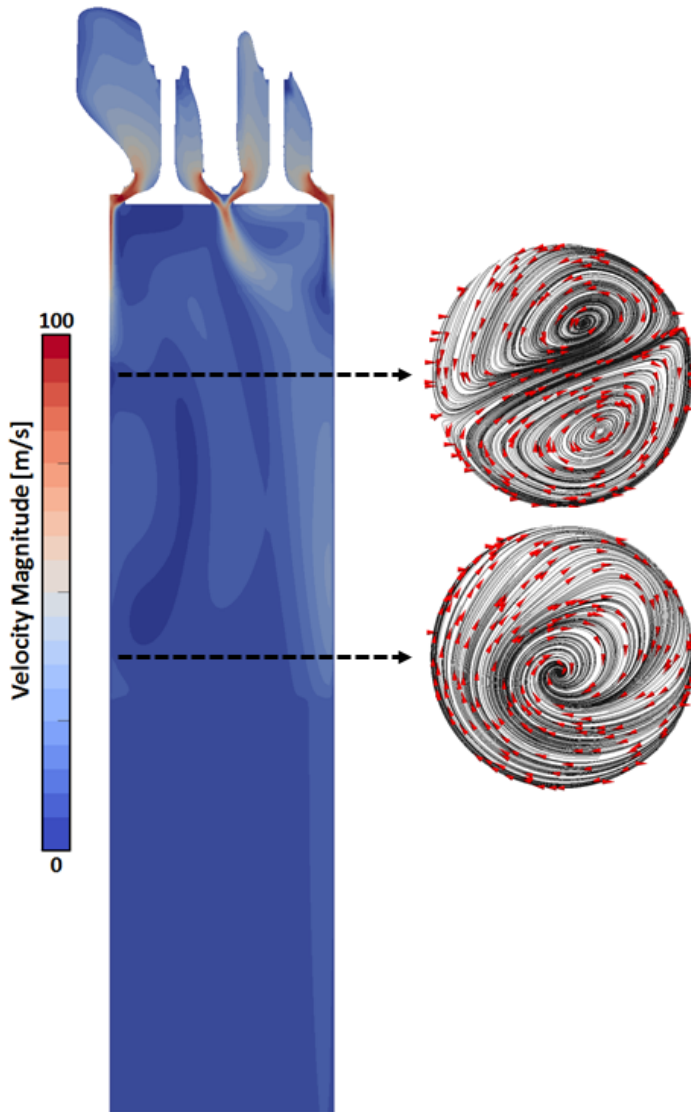


Figure 5.11: *Velocity magnitude map on the valves plane with planar streamlines taken on cutplanes perpendicular to cylinder axis. Distance from engine head: top - $0.8 \times \text{Bore}$, bottom - $1.6 \times \text{Bore}$.*

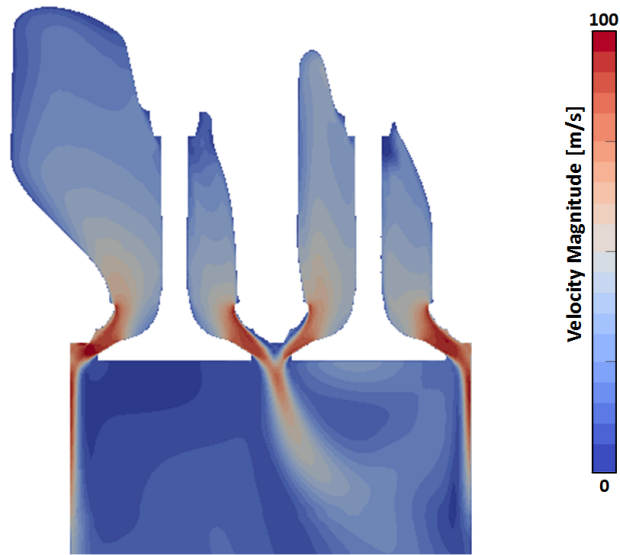


Figure 5.12: *Details of the velocity magnitude close to valve ports.*

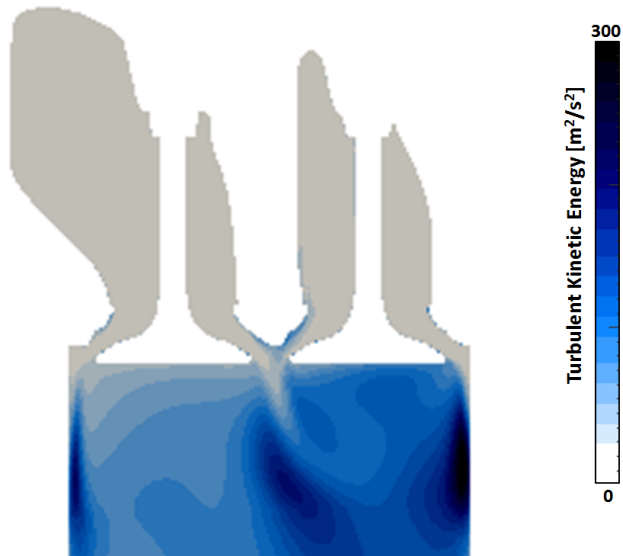


Figure 5.13: *Details of the turbulent kinetic energy close to valve ports.*

5.3 LES approach

The RANS analyses just presented have shown some deficiencies in swirl torque prediction. In order to investigate possible causes, the same system have been simulated adopting a LES approach and focusing on 3 different valve lifts:

- 2 *mm*, as representative of a lift with reasonable agreement between computational results and experiments,
- 5 *mm*, where the maximum relative error between computations end experiments occurs,
- 10 *mm*, where the maximum absolute error between computations end experiments occurs.

5.3.1 Case setup

The computational setup of the LES cases follows the one adopted for RANS, except for some aspects that will be now discussed.

Firstly, the mesh has been properly refined in order to have a reasonable grid density, with a bulk mesh size of 1 *mm* in the cylinder and reaching 1/2 of the bulk mesh in the valve region. In this case no wall layer has been extruded and the mesh has been isotropically refined up to 1/4 of the bulk size at solid surfaces. The meshes so obtained totaled nearly 7 M cells and a comparative view of a RANS mesh and LES mesh for the 5 *mm* valve lift is shown in Figure 5.14.

Numerical setup has been modified adopting a transient formulation, compulsory for LES. In particular a blended first/second order implicit scheme (Crank-Nicolson) has been chosen, as a reasonable trade-off between accuracy and stability. The advection scheme has been modified as well, opting for the second order TVD Gamma scheme [33]. The pressure-velocity coupling has been performed using the *PIMPLEC* algorithm ([38], [79]), described in Paragraph 2.5.6. The timestep has been set to 2×10^{-7} s, guaranteeing a maximum $Co < 2$ during the solution. The choice of exceeding the value of $Co = 0.5$, as suggested in Section 2.6, has been dictated by the need to contain the computation runtime. However, it has been verified that the number of cells in which $Co > 0.5$ is, on average, less than 0.1%.

On the ducts inlets a fixed total pressure condition is prescribed, while at the dummy cylinder outlet a wave-transmissive condition is set allowing proper treatment of the outflowing pressure waves. The far-field pressure is

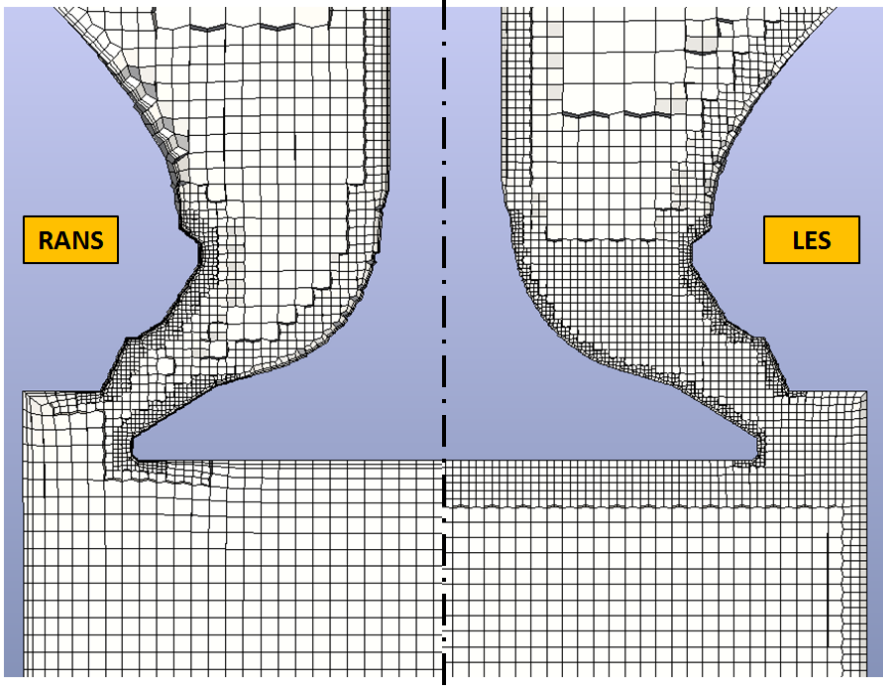


Figure 5.14: Grid comparison for the 5 mm valve lift. Left: RANS, right: LES.

set to the experimental nominal outlet value and an outflow convective condition is set for velocity. More details on these type of boundary conditions can be found in [38]. The velocity inflow boundary condition for the inlets is obtained from the patch-face normal component of the internal-cell value, as done for the RANS setup. No explicit synthetic turbulence has been fed at inflows, since a natural onset of turbulence is expected before reaching the valve. Despite the significant refinements applied to the walls, it has not been possible to reach y^+ values close to unity, therefore wall functions have been applied to all solid boundaries. Table 5.3 summarizes the boundary conditions applied for the LES analyses.

The LES setup has been completed opting for the WALE model as SGS model. The Sigma model, while being probably slightly superior in performance, is, at least in its actual non-optimized version, too expansive for the current analyses. The dynamic WALE has not been considered, having demonstrated a lesser accuracy than the (static) WALE model. To be noticed that being these simulation compressible, the WALE model here used has been properly re-implemented in a compressible formulation.

In order to save computing time, for each valve lift the domain has been

Boundary	Condition type	Variable	Condition
Inlets	Total pressure inflow	U	Pressure driven inflow
		p	Total pressure = 100800 Pa
		T	Static temperature = 298.15 K
Outlet	Wave transmissive outflow	U	Outflow
		p	$p_{\infty} = 95897$ Pa
		T	Outflow
Walls	No-slip wall	U	Wall function managed
		p	Zero gradient
		T	Zero gradient (Adiabatic)

Table 5.3: *Virtual flow bench boundary conditions (LES).*

initialized with the corresponding RANS results. The simulations have been then run for a total of 0.06 s and statistics have been accumulated after 0.025 s in order to allow the flow bench to reach a statistically steady regime.

5.3.2 Quality assessment

The analyses of results has been preceded by the quality assessment in analogy with the methodology followed for the test cases of Chapter 4. For sake of brevity, just the 5 mm valve lift case will be here discussed, since the other lifts exhibited a similar behavior.

The overall resolution reached in the fluid domain is reasonable. As shown in Figure 5.17, the M parameter is pretty low in the whole cylinder but high levels are reached in the ducts. In fact, as Figure 5.15 highlights, the resolution threshold is violated almost uniformly upstream the valves. Furthermore, in the former picture it is also possible to notice a weak spike at the beginning of the swirl meter, probably to be ascribed at the blocking effect of the porous region on the tangential velocity components.

Figure 5.18 reports the mean viscosity ratio map and it is evident the effect of the porous region. After an initial rise in SGS viscosity, the straightening of the flow greatly reduces the sub-grid activity. A peak in the SGS viscosity appears in the region where the two valve jets interact as shown in Figure 5.20. In the same picture it is also possible to notice the effect of grid density changes: as expected, the sub-grid activity tends to rise when the fluid encounters a cell size increase in the mesh.

Figure 5.16 shows the instantaneous y^+ distribution on the system. Most of the walls present $y^+ < 20$, while peak values up to 80 are reached in some portions of the head. These ranges are aligned to the ones obtained for the Thobois test case of Chapter 4, and represent a reasonable tradeoff between

accuracy and computational costs.

Finally, the mean SGS turbulent kinetic energy map is reported in Figure 5.19 where a close-up view of the valve region is presented. Here again it is shown that the jets interaction region is interested by a relevant sub-grid activity along with the liner near-wall region.

The quality assessment here presented has highlighted some domain regions of potential improvement in the turbulence resolution level. In particular, for the ducted region upstream the valves a local mesh refinement would be beneficial. Despite this, as the next paragraph will demonstrate, the predictive accuracy of the performed analyses is reasonable and more than sufficient for the purpose of the present investigation.

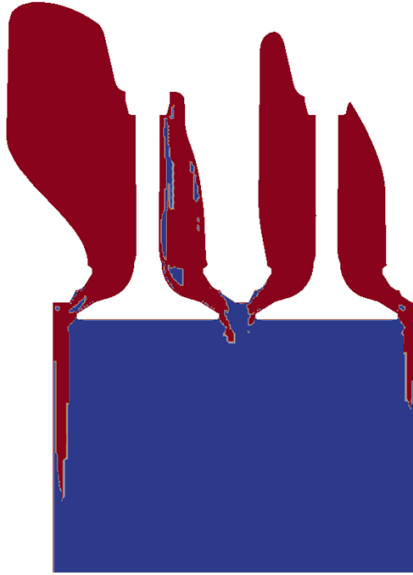


Figure 5.15: Details of the mean M parameter close to valve ports for valve lift 5 mm. In red the regions with $M > 0.2$.

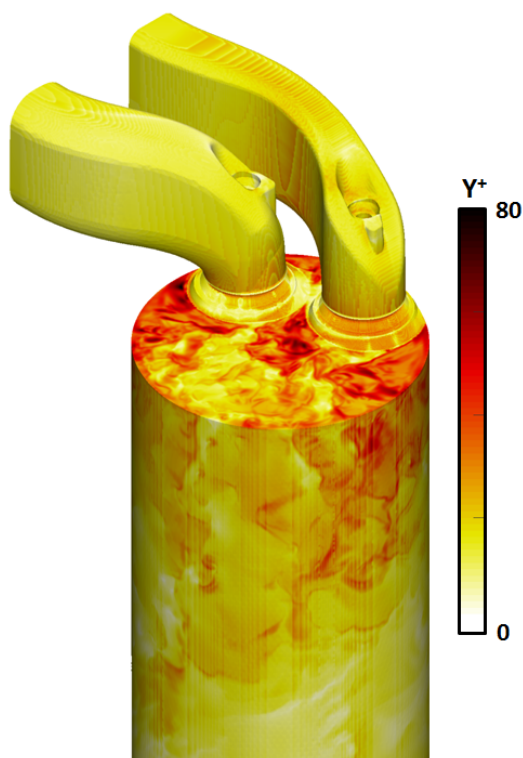


Figure 5.16: *Instantaneous y^+ map for the 5 mm valve lift.*

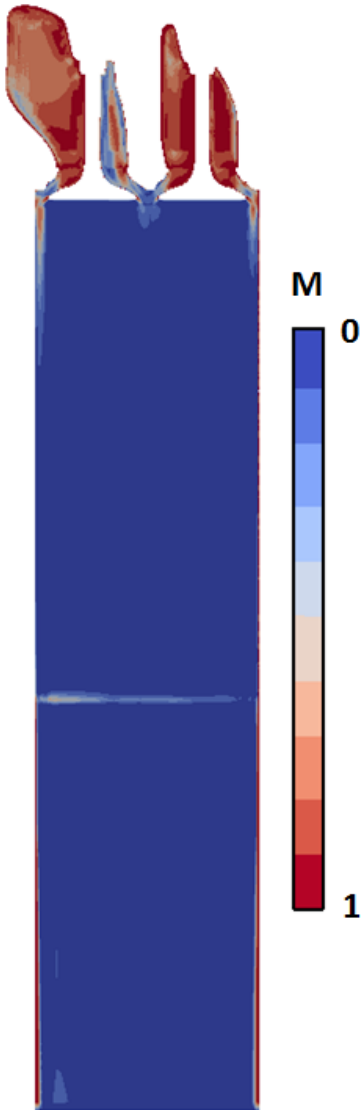


Figure 5.17: *M* parameter map on the valves plane. Valve lift 5 mm.

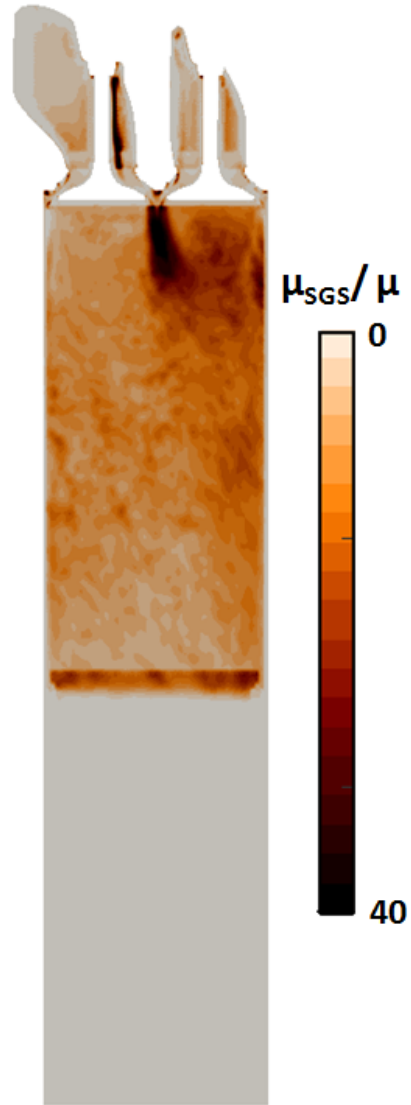


Figure 5.18: Viscosity ratio map on the valves plane. Valve lift 5 mm.

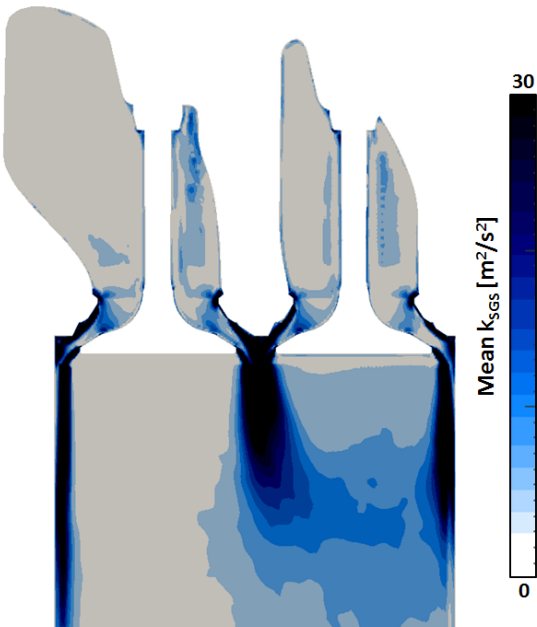


Figure 5.19: Details of the mean SGS turbulent kinetic energy close to valve ports for valve lift 5 mm.

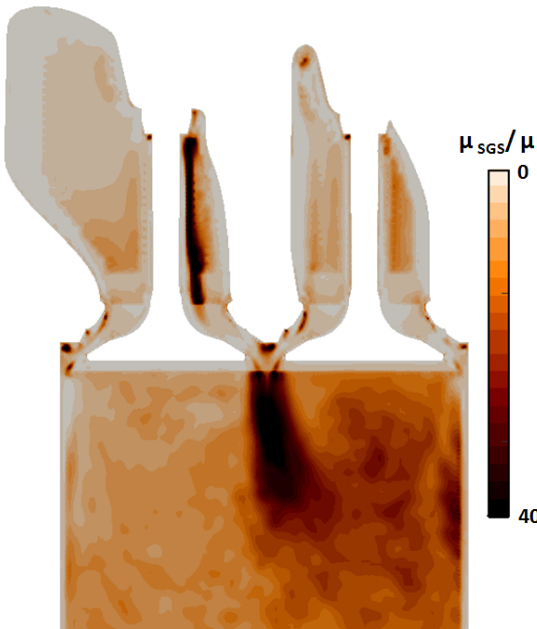


Figure 5.20: Details of the mean viscosity ratio close to valve ports for valve lift 5 mm.

5.3.3 Results

Results post-processing has firstly addressed mass flow rate and swirl torque evaluation. The procedure adopted is similar to the one used for RANS simulations, except that now, since dealing with transient simulations, proper time averaging has been performed. The analyses have been run for a total time of 0.06 s and the field averaging has been taken over the last 0.035 s , ensuring a statistically stationary fluid-dynamic regime for the sampling. A snippet of the mass flow rates evaluated at domain inflows and outflow is reported in Figure 5.21.

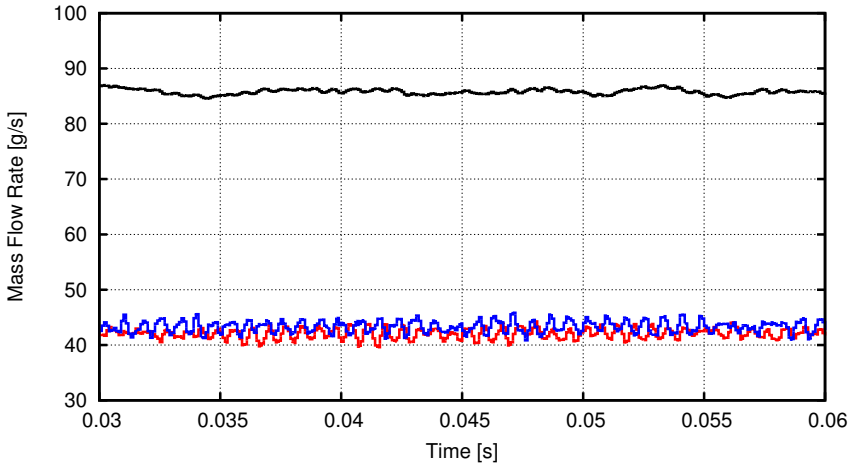


Figure 5.21: Mass flow rate vs time for the 10 mm valve lift: Outlet (black), Inlet 1 (red), Inlet 2 (blue).

Mass flow rates for the valve lifts of interest are reported in Figure 5.22 and in tabular format in Table 5.4, along with the values obtained from the previous analyses. The predictions are in good agreement with the experimental data and have comparable accuracy to the ones obtained from the RANS simulations.

Swirl torques are reported in Figure 5.23 and tabulated in Table 5.5. Here the LES results are in decent agreement with the experiments and definitely outperform RANS results.

Discrepancies between experiments and LES for both mass flow rate and swirl torque remain still sensible at low lift (2 mm). Possible improvements could be obtained by grid refinements in the valves region, where some resolution deficiencies have been highlighted by the quality assessment. However, it must be noted that while it is always advisable to reach the higher (reasonable) accuracy, from an industrial perspective, the results

accuracy related to low valve lifts has the least importance for Swirl Ratio evaluation purposes. In fact, these lifts are characterized by low flow rates and provide therefore marginal contributions to the in-cylinder angular momentum during intake stroke.

Instantaneous maps of the flow field are reported in Figures 5.24-5.27. For sake of simplicity, only the 5 mm case has been here reported, since the other lifts presented qualitatively similar distributions.

Figure 5.24 and Figure 5.25 show respectively the instantaneous velocity magnitude and the instantaneous absolute pressure fields. As expected by a LES simulation, small flow structures are visible, in particular close to the valves, where the valves jets interact. The chaotic nature of the flow is suggested also by Figure 5.27 in which the instantaneous viscosity ratio is reported.

The system pressure drop is almost entirely concentrated across the valve throat, where the strongest flow accelerations occur. It is also possible to notice the effect of the porous region which dumps out the perturbations. Such phenomenon is visible in Figure 5.26 too, in which the drop in the SGS turbulent kinetic energy is particularly manifest. Furthermore, these figures detail some of the flow characteristics typical of the device and already detected by the RANS simulations, such as the local fluid recirculations past the valves and the jets attachment to the liner walls after impingement.

Turbulent kinentic energy spectra evaluated in three different points are reported in Figure 5.29. The probes locations are shown in Figure 5.28. Probe A lies close to the valve guide of duct 1, probe B is located underneath valve 1 head and probe C is located in between the two valves, where strong jets interactions are expected.

For all the three points the resolution level seems decent. Point A exhibits the most noticeable deviation from the $-5/3$ theoretical slope, probably due to a insufficient resolution of the turbulent energy content in the ducts, as mentioned in Paragraph 5.3.2.

Point B is characterized by a pretty neat spectrum with a trend quite close to the the theoretical one. This point lies in a fluid recirculating region of the domain and this suggests that its turbulent kinetic energy content should be low-frequency biased, thus allowing an easier resolution of the spectrum respect to the other points.

Point C exhibits a resolution level in between the other points. Being positioned in the region where the valves jets collide and interact, it is characterized by strong temporal velocity gradients and high turbulent kinetic energy levels, as shown in Figure 5.19. Despite quality assessment ensured

proper resolution for this location, due to the local nature of the flow, it can be expected the turbulent kinetic energy content to be more spread over frequencies than point B, thus explaining the slightly higher deviation from the theoretical trend.

Figure 5.30 reports the spectral density of the pressure fluctuations for the same aforementioned points. The spectra are quite spread in terms of frequency distribution with greater contribution coming in the range of $300 \div 3000 \text{ Hz}$. Some noise is probably present here and, for a better spectral resolution, a longer time sampling would be beneficial. It must be noted, however, that low-frequency signals are always costly to properly resolve with LES, since they require a prohibitively extended simulated time, as explained in [59]. This is industrially (and academically) unfeasible and a reasonable compromise between accuracy and computational effort is usually made. Despite this, it is possible to notice a distinct peak in point B spectrum at nearly 1450 Hz . The same peak is visible in Figure 5.31 which reports the spectral density of the mass flow rate fluctuations at Inlet 1. Other signal peaks of lower intensity are present close to the 1450 Hz one. At present, no clear explanation can be given on the cause of such peaks. Speculatively, they could be associated to resonance frequencies of the duct, but further investigations would be needed to assess this hypothesis.

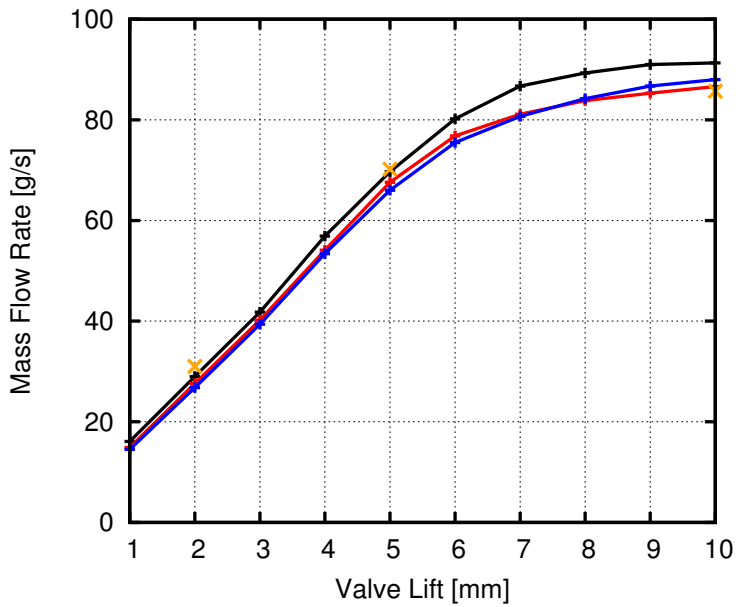


Figure 5.22: Mass flow rate as a function of valve lift: experiments (black), AVL Fire® RANS (red), OpenFOAM® RANS (blue), OpenFOAM® LES (orange).

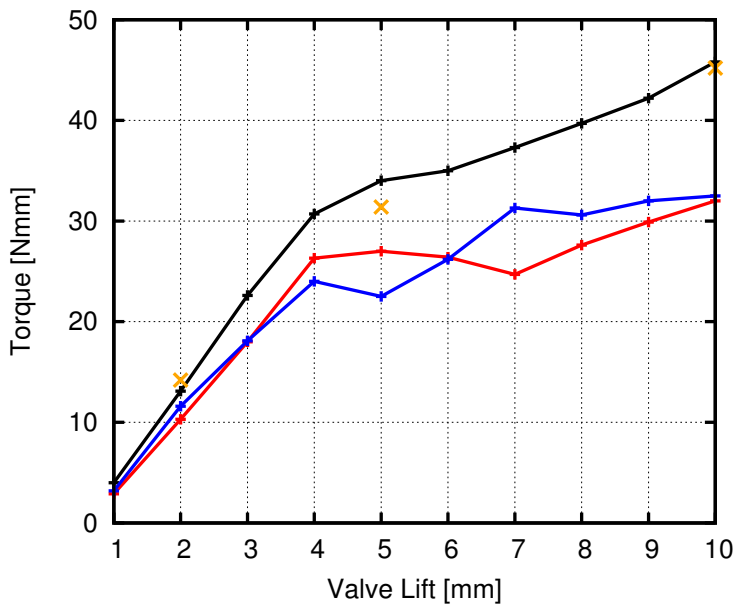


Figure 5.23: Swirl torque as a function of valve lift: experiments (black), AVL Fire® RANS (red), OpenFOAM® RANS (blue), OpenFOAM® LES (orange).

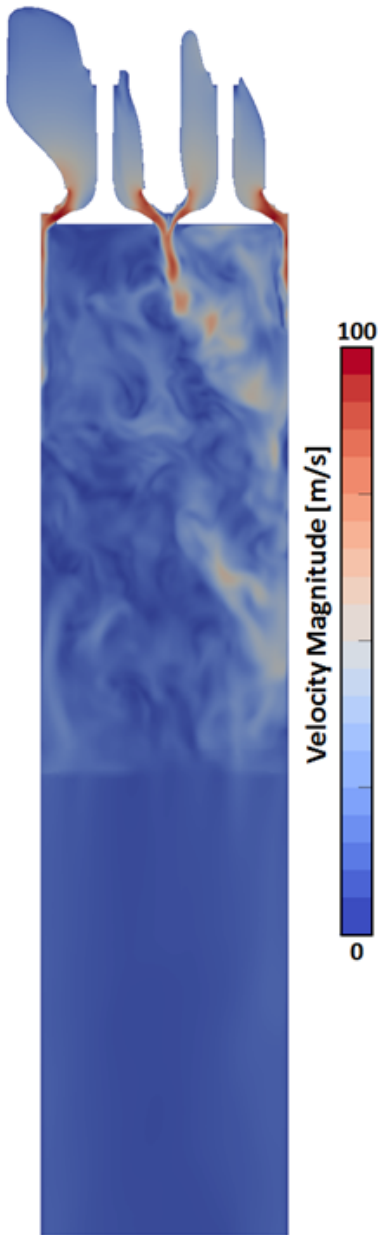


Figure 5.24: *Instantaneous velocity magnitude map for the 5 mm valve lift.*

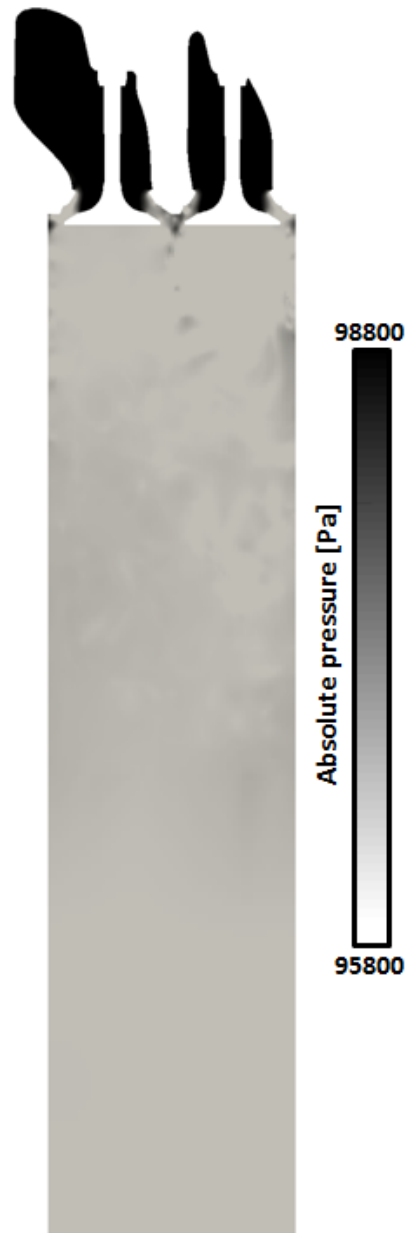


Figure 5.25: *Instantaneous absolute pressure map for the 5 mm valve lift.*

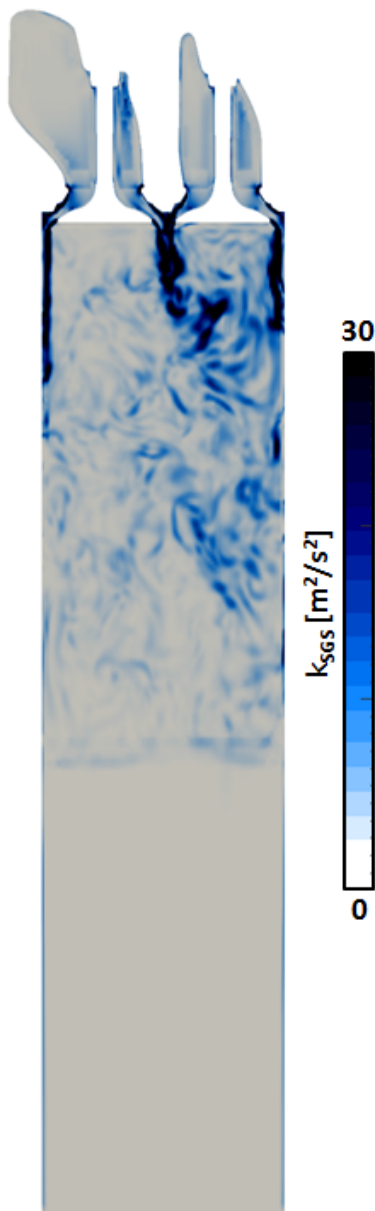


Figure 5.26: *Instantaneous SGS t.k.e. map for the 5 mm valve lift.*

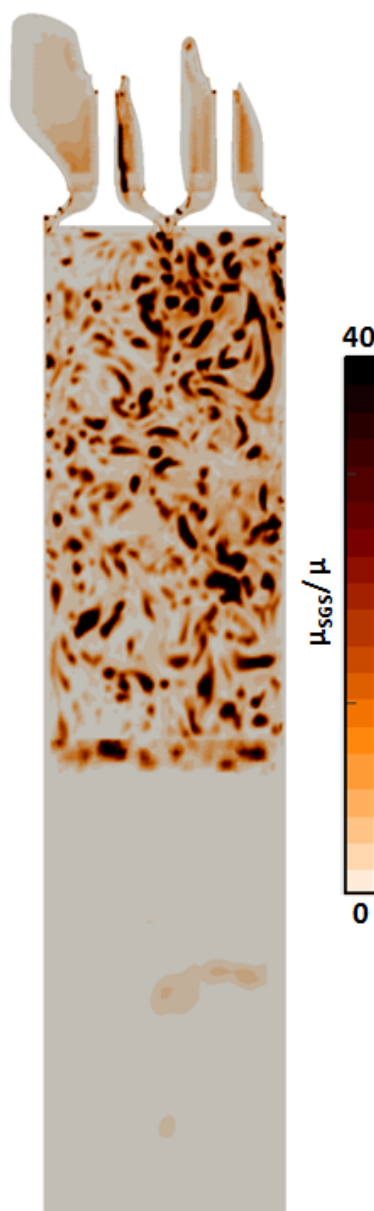


Figure 5.27: *Instantaneous viscosity ratio map for the 5 mm valve lift.*

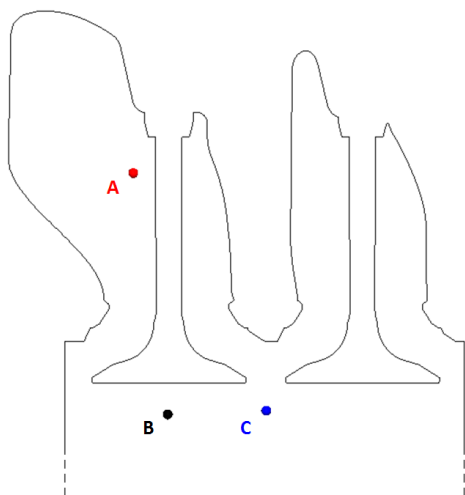


Figure 5.28: *Spectra probes locations.*

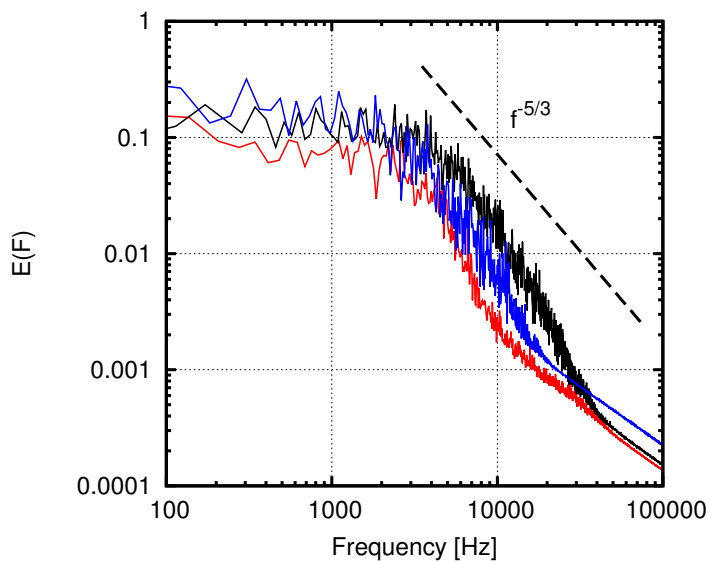


Figure 5.29: *Turbulent kinetic energy spectra for different locations: point A (red), point B (black), point C (blue).*

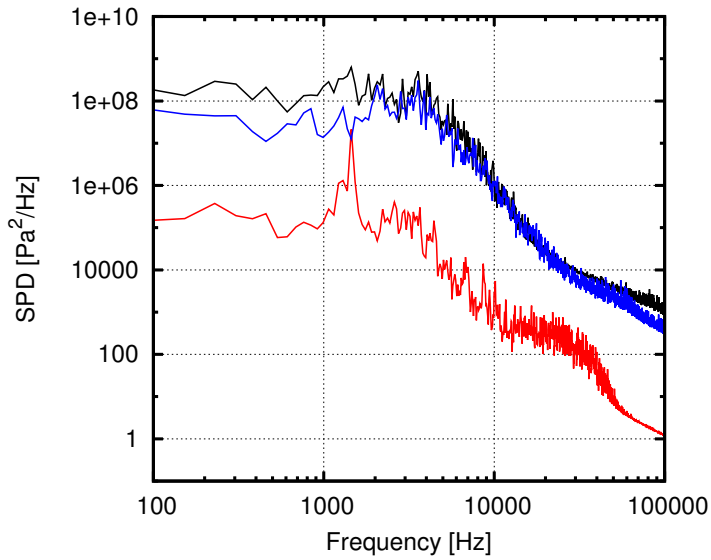


Figure 5.30: Spectral power density of pressure fluctuations for different locations: point A (red), point B (black), point C (blue).

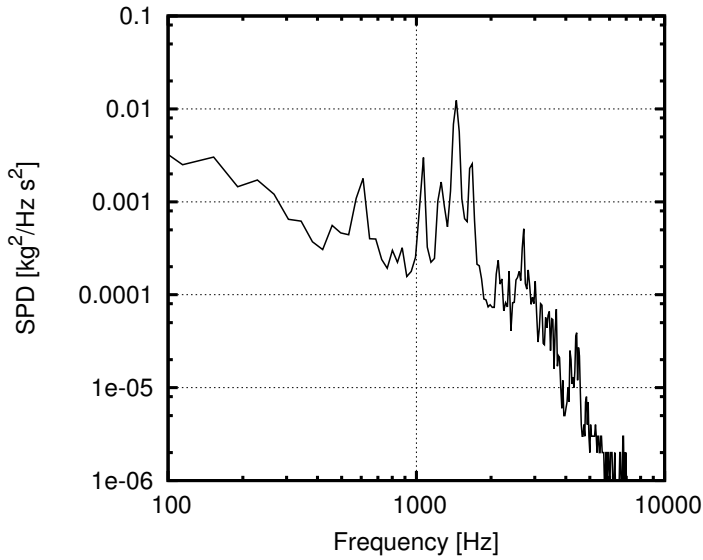


Figure 5.31: Spectral power density of mass flow rate fluctuations at Inlet 1.

In order to shed some light on the accuracy deficiencies of the swirl torques predicted by the initial RANS simulations, their flow fields have been compared to the ones obtained by the LES simulations. These latter analyses have in fact demonstrated a much better agreement with the experiments. Despite not claiming LES to have a fidelity comparable to a PIV or a LDV, they are anyway useful get some insights on the system fluid-dynamics providing full 3D flow fields and to highlight potential causes of the aforementioned deficiencies.

The post-processing topology employed for this investigation is shown in Figure 5.32. Planar streamlines are generated on three sampling planes perpendicular to cylinder axis and placed respectively at $0.8 \times \text{Bore}$, $1.2 \times \text{Bore}$ and $1.6 \times \text{Bore}$ far from the engine head. These streamlines allow the visualization of the vortical swirling structures that develop inside the cylinder. Moreover, velocity vector heads created on the same planes help in identifying rotation directions. All the samplings are presented accordingly to a top-to-bottom view (as indicated in Figure 5.32) and have been generated using the mean velocity field.

Figure 5.34 reports the streamlines obtained for the 2 mm valve lift. Close to the valves the fluid motion presents the main clockwise rotating vortex promoted by the ducts arrangement and a small counter-rotating one. This latter quickly dissipates while descending along the cylinder and is not detectable anymore on the mid plane. The main vortex has anyway continued its development and its rotation axis is now better centered respect to the cylinder axis. The bottom plane reveals the presence of the main clockwise rotating vortex only, whose rotation center remains off-axis. The fluid-dynamic evolution of the vortical structures for the 2 mm lift does not exhibit significant differences between RANS and LES. This was somewhat expected since, for such lift, the predictive accuracy of the two approaches are similar.

The situation is quite different for the 5 mm valve lift, reported in Figure 5.34. Now the cylinder top slice presents two distinct counter rotating vortexes of similar extent, schematically sketched in Figure 5.33. Proceeding to the mid section, while in LES results only the main clockwise rotating vortex is present, in RANS results the secondary counter-rotating vortex still exists. Such vortex, despite its small size and its weakness, strongly affects the development of the main swirling structure, which is still squished in nearly half of the cylinder section.

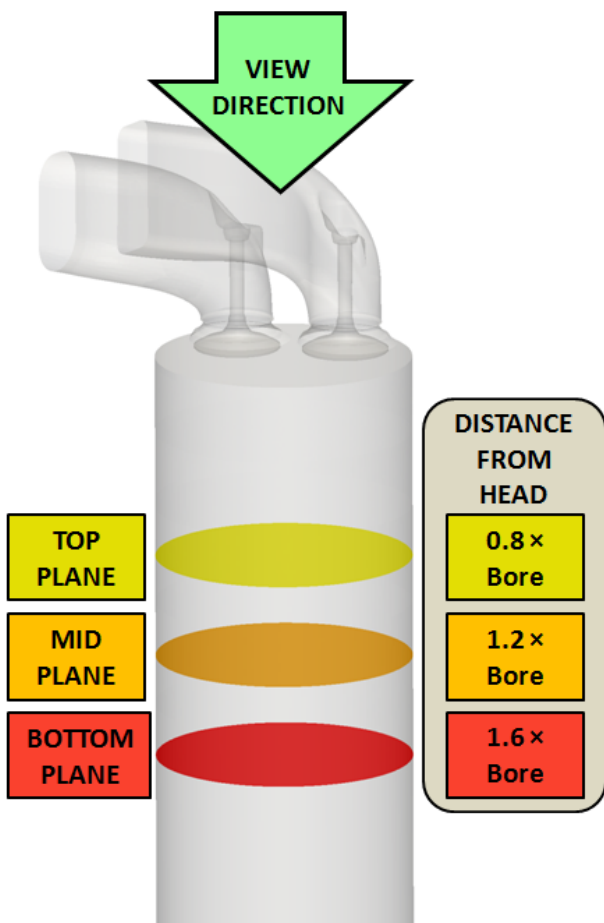


Figure 5.32: *Post-processing topology for planar streamlines.*

The bottom plane reveals a well-shaped and centered main vortex for LES, while, for RANS, despite the absence of secondary structures, the main vortex remains significantly off-axis. Moreover, for the latter case it can be speculatively presumed that the prolonged interaction of the two counter-rotating vortexes has dissipated a greater fraction of the swirling strength of the main vortex respect to the LES case.

Finally, it must be noted that a departure of the vortical structure axis from the cylinder axis, which coincide with the swirl meter axis, will result in a lower swirl torque, as can be inferred by the torque evaluation methodology, reported in Appendix B.

The 10 *mm* valve lift, here not reported, exhibits results qualitatively similar to the 5 *mm* ones.

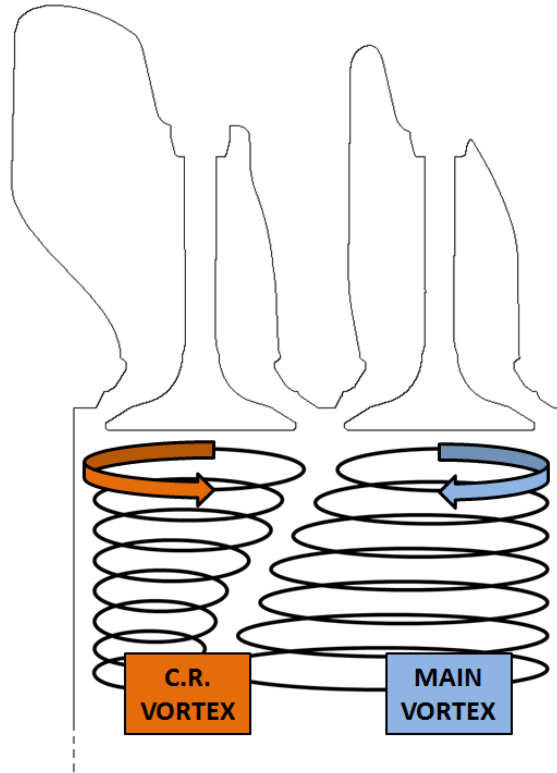


Figure 5.33: *Sketch of vortexes development inside the cylinder.*

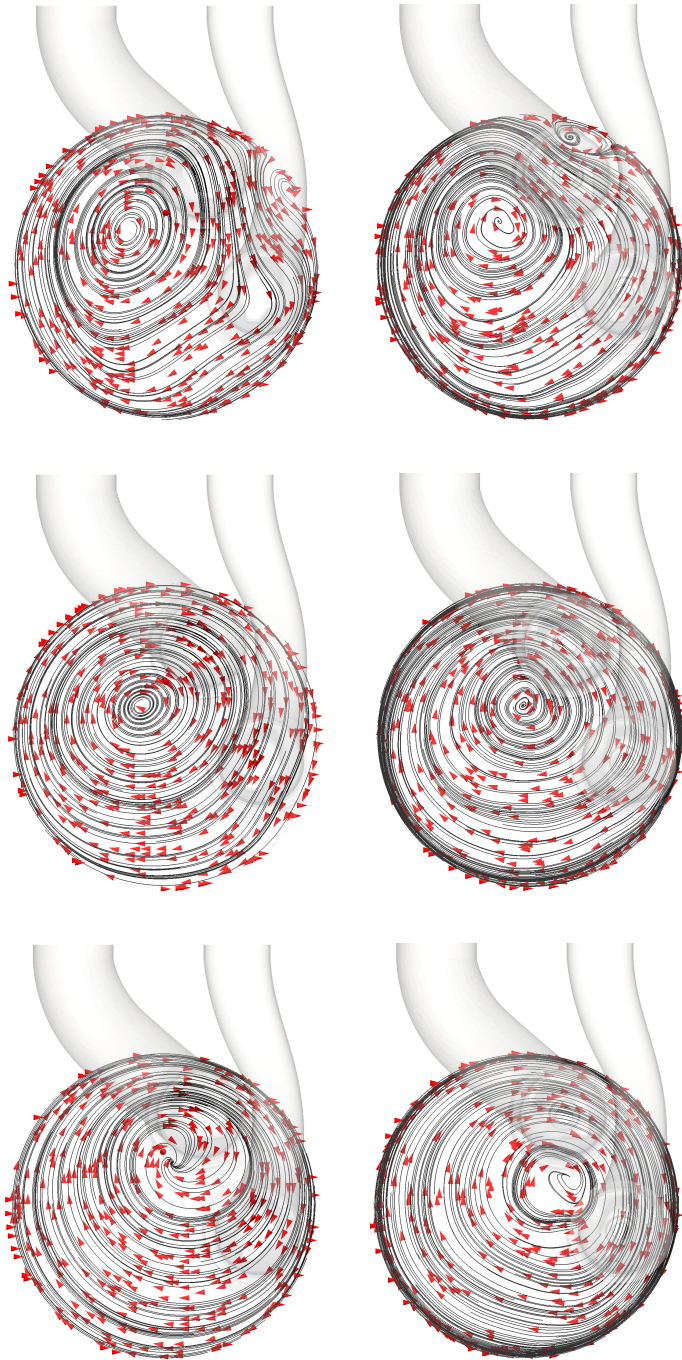


Figure 5.34: Planar streamlines generated on post-processing topology of Figure 5.32. Valve lift 2 mm. Left column: RANS, right column: LES. First row: TOP planes, second row: MID planes, third row: BOTTOM planes.

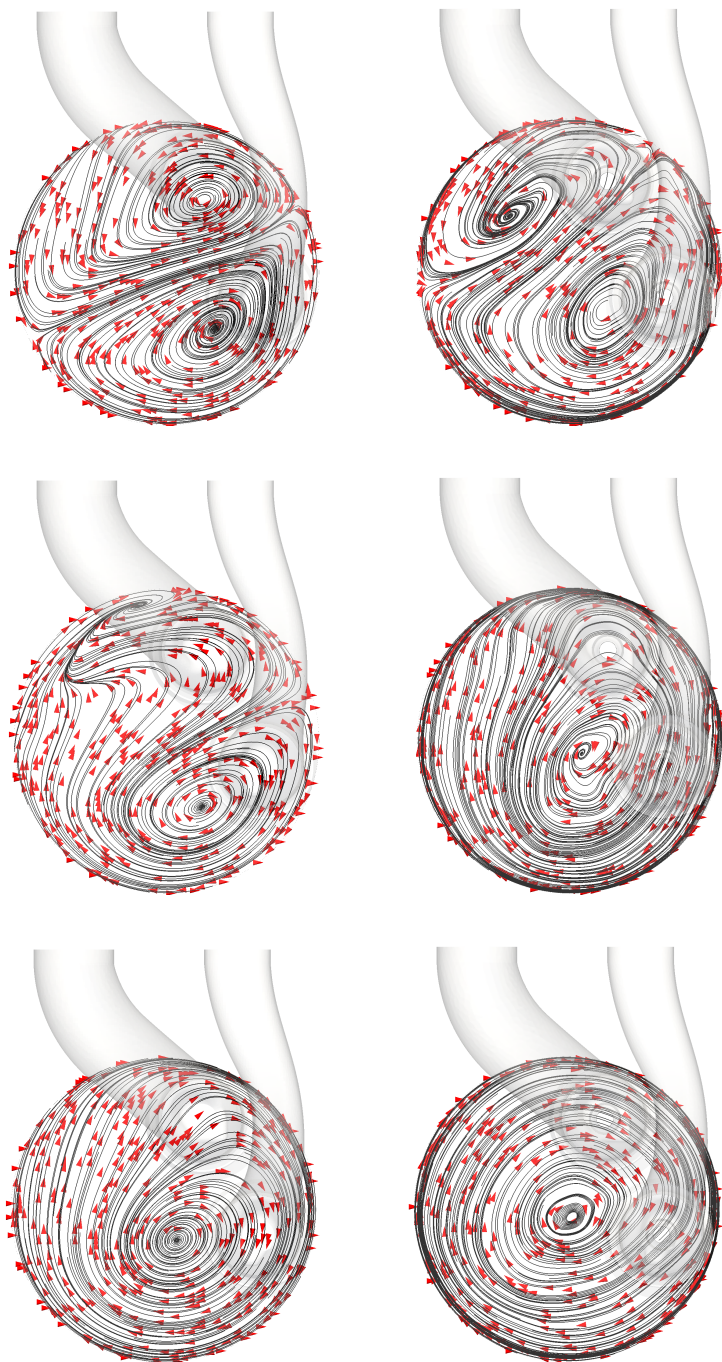


Figure 5.35: Planar streamlines generated on post-processing topology of Figure 5.32. Valve lift 5 mm. Left column: RANS, right column: LES. First row: TOP planes, second row: MID planes, third row: BOTTOM planes.

5.4 RSTM RANS approach

The results comparison of the previous paragraph has suggested that reliable and accurate predictions for a virtual flow bench can be obtained only by proper resolution of the swirling structures which develop inside the system.

Despite the LES modeling fulfills such need, its cost is not negligible for a routine industrial usage or for the application within optimization loops. In such cases, a potential improvement over the standard RANS eddy-viscosity modeling, presented at the beginning of the chapter, could be obtained adopting a Reynolds Stress Tensor Model (RSTM). This class of RANS turbulence models does not rely on the eddy-viscosity hypothesis, but rather tries to directly model the Reynolds Stress Tensor [10].

Due to its formulation, the RSTM has intrinsic capabilities to resolve flow anisotropic features and curvature effects typical of vortical structures and could therefore bring improvements for virtual flow bench applications. On the other hand, while the commonly adopted RANS models, such as the $k-\varepsilon$ or the $k-\omega$ and their several variants, require the solution of two transport equations for the turbulent quantities, the RSTM approach requires the solution of seven additional transport equations [10].

The RSTM RANS approach is therefore more expensive than an eddy-viscosity based RANS approach, with an extra computational cost that could be estimated in an increase of the wall-clock time between 40÷50%. Nevertheless, they remain much cheaper than LES and still affordable for routine industrial simulations.

In order to assess the effectiveness of such approach, the cases have been run using the Launder-Rodi-Reece (LRR) RSTM model [80], already available in the standard libraries of *OpenFOAM*[®]. All the remaining setup and boundary conditions have been the same as the ones described in Section 5.2.

Results in terms of mass flow rate and swirl torque for the various lifts are shown in Figures 5.36 and 5.37 and summarized in Tables 5.4 and 5.5.

While mass flow rates predictions are aligned to the previous results, a definitely better agreement with experiments has been obtained for the swirl torques. For these latter, the deviations are contained in a 10÷15% range, with a maximum relative error of 17.5% for the 1 mm valve lift.

The increased predictive accuracy obtained with the LRR model is confirmed by the streamlines comparison proposed in Figure 5.38. Here, it is evident how the RSTM vortical flow patterns are much closer to LES than the original eddy-viscosity RANS ones.

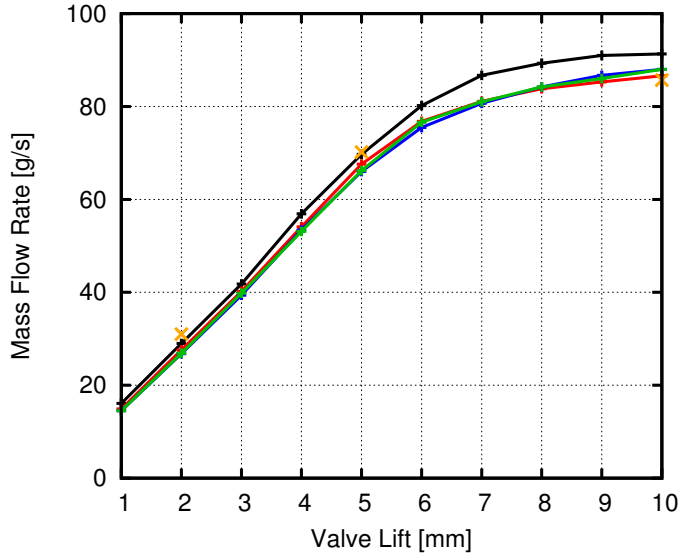


Figure 5.36: Mass flow rate as a function of valve lift: experiments (black), AVL Fire® Eddy-viscosity RANS (red), OpenFOAM® Eddy-viscosity RANS (blue), OpenFOAM® RSTM RANS (green), OpenFOAM® LES (orange).

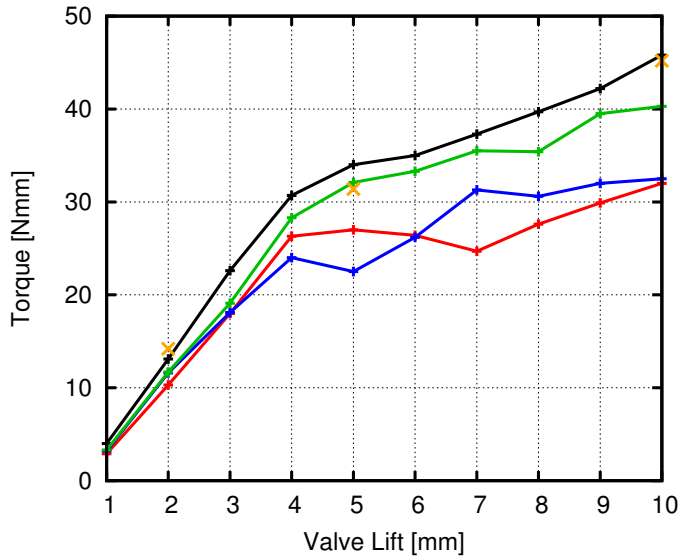


Figure 5.37: Swirl torque as a function of valve lift: experiments (black), AVL Fire® Eddy-viscosity RANS (red), OpenFOAM® Eddy-viscosity RANS (blue), OpenFOAM® RSTM RANS (green), OpenFOAM® LES (orange).

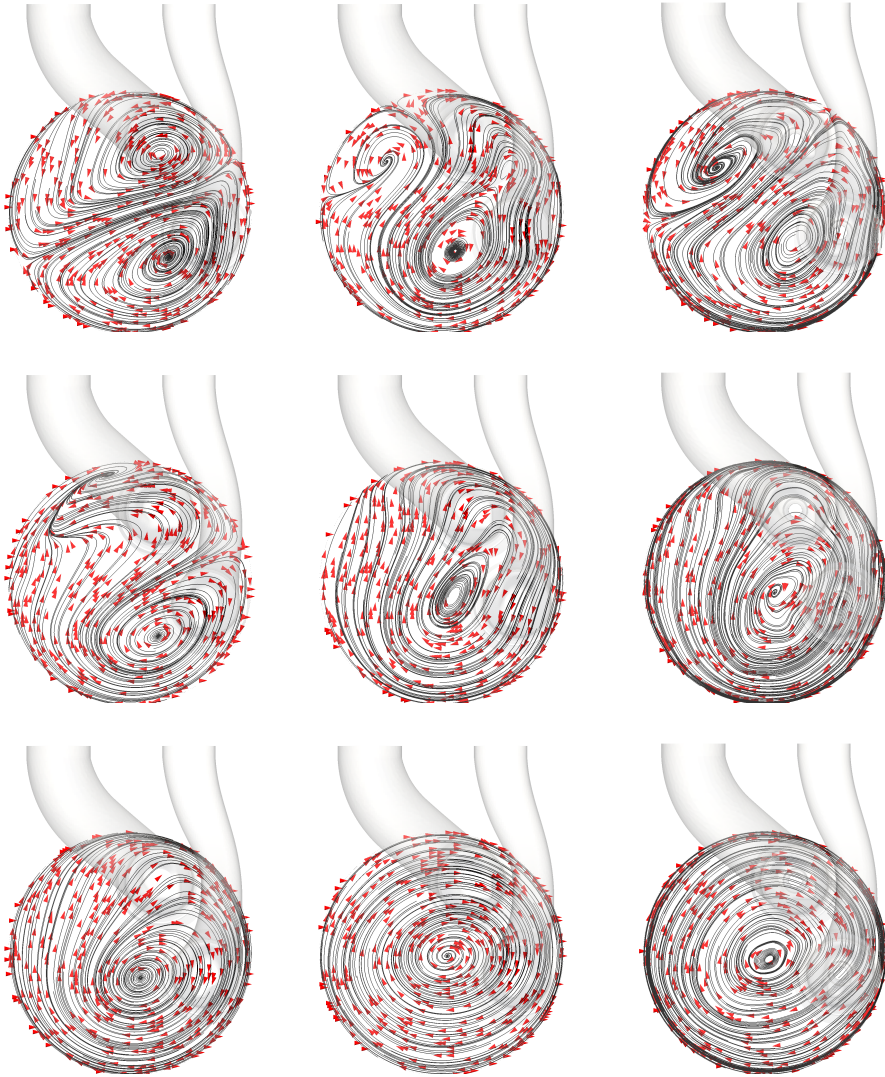


Figure 5.38: *Planar streamlines generated on post-processing topology of Figure 5.32. Valve lift 5 mm. Left column: Eddy-viscosity RANS, middle column: RSTM RANS, top column: LES. First row: TOP planes, second row: MID planes, third row: BOTTOM planes.*

5.5 Summary of results

The following tables summarize the results obtained for the various analyses presented in this chapter.

Lift	Exp.	AVL-FIRE (Eddy-visc. RANS)		OpenFOAM (Eddy-visc. RANS)		OpenFOAM (RSTM RANS)		OpenFOAM (LES)	
[mm]	Value [g/s]	Value [g/s]	Δ % (vs Exp.)	Value [g/s]	Δ % (vs Exp.)	Value [g/s]	Δ % (vs Exp.)	Value [g/s]	Δ % (vs Exp.)
1	16.1	14.9	-7.4	14.5	-10.0	14.5	-9.9	-	-
2	29.0	27.6	-4.7	26.8	-7.6	26.9	-7.2	31.0	6.9
3	41.8	40.2	-3.9	39.4	-5.9	39.8	-4.8	-	-
4	56.9	54.1	-4.9	53.4	-6.1	53.1	-6.6	-	-
5	69.7	67.6	-3.0	66.0	-5.3	66.2	-5.0	70.2	0.7
6	80.2	76.8	-4.2	75.5	-5.8	76.6	-4.5	-	-
7	86.7	81.1	-6.4	80.7	-6.9	81.0	-6.6	-	-
8	89.3	83.8	-6.1	84.2	-5.7	84.2	-5.7	-	-
9	91.0	85.3	-6.2	86.7	-4.7	86.0	-5.5	-	-
10	91.3	86.6	-5.1	88.0	-3.6	88.0	-3.6	85.7	-6.1

Table 5.4: Steady flow bench mass flow rates.

Lift	Exp.	AVL-FIRE (Eddy-visc. RANS)		OpenFOAM (Eddy-visc. RANS)		OpenFOAM (RSTM RANS)		OpenFOAM (LES)	
[mm]	Value [Nmm]	Value [Nmm]	Δ % (vs Exp.)	Value [Nmm]	Δ % (vs Exp.)	Value [Nmm]	Δ % (vs Exp.)	Value [Nmm]	Δ % (vs Exp.)
1	4.0	2.9	-28.0	3.2	-20.8	3.3	-17.5	-	-
2	13.1	10.3	-21.4	11.6	-11.5	11.7	-10.8	14.2	8.3
3	22.6	18.0	-20.6	18.1	-19.9	19.1	-15.6	-	-
4	30.7	26.3	-14.3	24.0	-21.9	28.3	-7.8	-	-
5	34.0	27.0	-20.6	22.5	-34.0	32.1	-5.6	31.4	-7.7
6	35.0	26.4	-24.6	26.2	-25.0	33.3	-4.8	-	-
7	37.3	24.7	-33.8	31.3	-16.0	35.5	-4.7	-	-
8	39.7	27.6	-30.6	30.6	-23.1	35.4	-10.9	-	-
9	42.2	29.9	-29.2	32.0	-24.2	39.5	-6.4	-	-
10	45.8	32.0	-30.1	32.5	-29.0	40.3	-12.0	45.2	-1.3

Table 5.5: Steady flow bench swirl torques.

5.6 Closure

The present chapter has presented an application of the LES simulation methodology to a steady flow bench for engine head performance evaluation. The goal of the study has been twofold: gain better insights of the system fluid dynamics and investigate the potential causes of predictive deficiencies emerged when a standard RANS simulation approach is used.

Firstly, RANS analyses of the head under investigation have been performed. While the system mass flow rates have been decently predicted, swirl torques exhibited severe discrepancies (up to $\sim 30\%$) respect to experiments. Similar findings have been obtained also using the commercial CFD code *AVL Fire*[®].

LES simulations have been then carried out for selected valve lifts. Quality assessment has been performed on the results ensuring that a reasonable resolution level has been reached. The predictions of both mass flow rates and swirl torques have presented a general good agreement with the experimental data. The evaluations of the LES computed flow fields allowed to highlight some characteristic system flow features such as recirculating regions, flow detachments and attachment to the liner walls.

Then, the LES predicted flow fields have been compared to the RANS ones. For some of the investigated valve lifts, the development of the vortical flow structures differed between the two approaches. In particular, the RANS cases that presented the greater discrepancies with the LES simulations exhibited a prolonged persistence of a secondary counter-rotating vortex in the cylinder. The interaction between such vortex and the main one resulted in a significant misalignment of the former with respect to the cylinder axis in proximity of the swirl meter. Furthermore, the prolonged persistence of the aforementioned interaction presumably contributed in a reduction of the main vortex swirling strength.

Finally, the insights gained with the LES simulations has suggested that a cause in the lack of accuracy on swirl torque predictions of the RANS simulations had to be ascribed to the poor resolution of the vortical flow structures developing in the cylinder. In order to improve RANS simulations accuracy, the cases have been run again employing a RSTM RANS model rather than the original eddy-viscosity *RNG* – $k\varepsilon$ turbulence model. RSTM models, particularly suited for swirling flows and sensible to streamlines curvature, allowed a much better agreement with both the LES simulations and the experimental data.

The conclusions that can be drawn from this study can be then summarized as follow:

- ***OpenFOAM*[®] predictive performance respect to commercial codes**

The comparison of *OpenFOAM*[®] with the commercial code *AVL Fire*[®], one of the most widely adopted CFD packages by engine manufacturers, resulted in a substantial equivalence in terms of predictive accuracy. This finding opens interesting cost-saving scenarios with the potential substitution of commercial packages with open-source software for engine air induction systems design and optimization.

- **LES flow details resolution capabilities**

The LES methodology applied to steady flow benches allows the accurate resolution of the fluid dynamic behavior characterizing these devices. Flow detachments in the valve seats, flow attachment to head and liner walls, valves jets dynamic interactions are just some of the fluid dynamic features that is possible to study in detail. Without claiming to have a degree of fidelity comparable to experiments, it can be used in substitution of bench data, when this is missing, or as a valuable integration to the experimental practice, providing full 3D details of the flow field.

- **Careful selection of RANS turbulence models**

The common eddy-viscosity RANS methodology applied to steady flow bench simulation could sometimes return significant errors in the swirl torque predictions. Such poor accuracy, not uncommon in the scientific literature, could compromise development and optimization processes of engine heads and air induction systems. A viable solution to overcome these predictive issues while still adopting an inexpensive steady-state RANS modeling approach is the choice of turbulence models sensible to streamline curvature, such as a Reynolds Stress Transport Model (RSTM).

A final note relates to the computational cost of the analyses performed in this chapter. Obviously, the steady-state RANS analyses are much cheaper than the LES simulations. To give a rough idea of this, the simulation cost for a single valve lift can be estimated to be:

- Eddy-viscosity RANS: 1 day on a 2 x Intel Xeon E5-2609 @ 2.40GHz workstation (8 cores total). Approximately 3 GB RAM needed.
- RSTM RANS: 1.5 days on a 2 x Intel Xeon E5-2609 @ 2.40GHz workstation (8 cores total). Approximately 3.5 GB RAM needed.
- LES: 35 days on a 4 x AMD Opteron 6212 @ 2.60GHz blade (32 cores total). Approximately 15 GB RAM needed.

Moreover, it has to be noted that the RANS simulations could be further sped-up by using an appropriate solution steering approach (Eulerian initialization, numerical schemes steering, etc.).

CHAPTER 6

Summary and Outlook

6.1 Research summary

The objective of the Doctoral research activity here presented has been to define and assess a suitable LES numerical methodology for ICE flows applications adopting open-source software components.

An evaluation of the open-source CFD codes panorama has led in choosing *OpenFOAM*[®] as the code of choice for the present work. In fact, it presented a reasonable maturity for the industrial applications of interest and its modularity allowed easy code extension/integration.

Numerical dissipation and dispersion of the code have been studied through an inviscid vortex convection test. The outcomes of such test have confirmed the capabilities of *OpenFOAM*[®] for LES applications provided that care is taken for space and time discretization.

Particular attention has been given in the evaluation of the simulations quality in terms of LES resolution capabilities. This has been done by firstly studying LES quality estimators and then selecting the ones better suited for ICE applications, which have been then used in the subsequent analyses.

The application of the quality parameters to the various simulations car-

ried out in this work allowed to conclude that, even if not suitable for results validation, they could be usefully employed to evaluate the optimality of the grid respect to the present flow, identifying regions of possible deficiencies in energy resolution.

One of the key components of the LES is the sub-grid scale model. While *OpenFOAM*[®] offered some standard SGS models in its default libraries, other models, better suited for ICE flows were needed. Among several options, the WALE, the dynamic (homogeneous) WALE and the Sigma models have been chosen and have been implemented in the code developing an ad-hoc library. While the WALE model could be considered a basic good choice for engine flow simulations, the other two represent recent evolutions and improvements of the former.

The SGS models so implemented have been tested using two different test cases: an abrupt expansion of a swirling flow (Dellenback test case) and the flow past a poppet valve in a steady flow bench (Thobois test case). For both test cases the predictive performance of the aforementioned models have been compared to the one obtained with the dynamic Smagorinsky model available in *OpenFOAM*[®] standard libraries and considered as a base reference for the assessment.

The outcome of the tests have shown that while all the three implemented models outperform the reference one, the WALE model in its static version is probably still the best choice. The dynamic WALE has in fact demonstrated some predictive deficiencies, while the Sigma model, despite being the most accurate of the four models tested, is also, by far, the most computationally expensive.

Other outcomes of the tests related to the discretization approach. One of the most peculiar results has been that the relation between grid density and accuracy remains unclear: in some cases employing a finer grid doesn't lead to accuracy improvements. This behavior has been imputed to the models constants which would need further tuning, eventually recurring to a dynamic localization procedure. A comparison of the resolution performance of hexahedral and tetrahedral elements confirmed the widespread idea of the superiority of the former on the latter. Tetrahedral elements, despite often returning field averages comparable to the hexahedral ones, tend to underestimate fluctuations as clearly visible in their dumped energy spectra. In the Thobois test case the orientation of the cells respect to the valve jet direction has been evaluated. The outcome of this further test led to conclude that employing a flow-aligned discretization doesn't necessarily improve the simulation accuracy. It is commonly assumed that cells oriented accordingly to the flow direction provide better results even

if guaranteeing such an orientation is not always practically feasible. The test provided therefore encouraging results towards the use of unstructured automatic hexahedral mesh generators, which are nowadays always more common CFD pre-processing tools.

The developed methodology and the knowledge acquired have than allowed to tackle a more realistic case: the LES approach has been used to investigate the fluid-dynamics of a Diesel engine intake system through a virtual flow bench. RANS analyses of such components presented poor agreement with experimental results and LES has been used as an investigation tool to shed some light over these predictive deficiencies.

The Diesel engine head under investigation has been firstly simulated using a standard eddy-viscosity RANS approach and the results have been compared to the ones obtained with a commercial code of reference (*AVL Fire*[®]) and to the experimental data kindly provided by the engine manufacturer. While the system mass flow rates for the different valve lifts have been decently predicted, swirl torques exhibited severe discrepancies (up to $\sim 30\%$) respect to experiments. The comparison of *OpenFOAM*[®] with *AVL Fire*[®], one of the most widespread CFD codes in automotive industry, resulted in a substantial equivalence in terms of predictive accuracy. This outcome opens interesting cost-saving scenarios with the potential substitution of commercial packages with open-source software for the analysis of engine air induction systems.

Afterwards, LES simulations have been carried out for the same system. In order to ensure that the simulations reached a reasonable resolution level, quality assessment using the parameters described in Chapter 3 has been carried out. The predictions of both mass flow rates and swirl torques have presented a general good agreement with respect to the experimental data, demonstrating the superior capabilities of LES over RANS in terms of simulation accuracy and reliability. The evaluations of the LES computed flow fields allowed to highlight some characteristic fluid-dynamic features such as flow detachments in the valve seats, flow attachment to head and liner walls and valves jets dynamic interactions.

LES predicted flow fields have been then compared to the RANS ones, using the former as a surrogate of experimental 3D data.

For some of the investigated valves lifts, the development of the vortical flow structures differed between the two approaches. In particular, the RANS cases that presented the greater discrepancies with the experiments, exhibited a prolonged persistence of a secondary counter-rotating vortex within the cylinder with respect to LES predictions. The interaction between such vortex and the main one resulted in a significant misalignment

of the latter with respect to the cylinder axis in proximity of the swirl meter. Moreover, the prolonged persistence of the aforementioned interaction presumably contributed in a reduction of the main vortex swirling strength, negatively affecting torque readings.

In order to overcome or, at least, to alleviate the highlighted predictive deficiencies while still adopting an inexpensive RANS modeling approach, a RSTM RANS model has been then tested. This class of turbulence models are better suited to resolve flow anisotropic features and streamlines curvature effects than the common eddy-viscosity RANS models, such as the RNG $k-\varepsilon$ initially adopted. The results demonstrated the superior performance of the alternative approach with predictions in much better agreement with experimental data and with computed flow fields pretty close to the LES ones.

Despite not being (yet) suited for routine industrial usage, LES has demonstrated to be a valuable tool for studying engine air intake systems. Without claiming to have comparable fidelity to experiments, it can be used in substitution of PIV or LDV measurements when these are not available or as a valuable integration capable of providing full 3D flow fields of the device under investigation. The level of detail provided by this computational technique could help engine designers in better understanding the fluid-dynamic behavior of the system with clear advantages for performance improvement and optimization.

6.2 Ongoing activities

Presently (March 2016) the LES analysis methodology here developed is going to be extended to dynamic applications in order to address flows of engines in operating conditions, firstly tackling motored configurations. The Sandia TCC engine [48] will to be simulated with the main target being the evaluation of cycle-to-cycle variations.

The methodology under development is based on the mesh target approach: the whole engine cycle is simulated using a set of mesh, each one valid within a certain angular CA range. During the simulation, the current mesh will be distorted in order to accommodate the motion of parts of the engine (i.e. valves, piston). The aforementioned distortion negatively affects the mesh quality which degrades progressively. When the quality falls below a certain user-defined threshold, the whole mesh is substituted with a new one valid for the subsequent CA range. While the substitution is performed, the flow fields are interpolated onto the new mesh and the solution process restarts.

In order to ease and automate the simulation process, a specific Python scripting package is currently under development. Functionally, it will cover the whole simulation workflow, automating target grids generation and computational setup, managing the runs and finally post-processing results to generate flow data statistics.

6.3 Future work

Beyond the completion of the TCC engine case mentioned in the previous section, possible areas of interest for future work could be the following:

- Localization of the dynamic Wale model. The implementation used in the present work adopts a homogeneous dynamic procedure for computing the model constant. An interesting alternative would be the adoption of a lagrangian localized dynamic procedure following the methodology described in [81]. This could potentially allow to improve the model's accuracy without severely affecting its (low) computational cost.
- Implementation of an Adaptive Mesh Refinement (AMR) procedure to automatically refine LES grids. The AMR algorithm would be driven by one of the LES quality parameters discussed in the thesis with the purpose to dynamically adapt the computational grid where the resolution is not adequate. This approach could therefore potentially alleviate the need for manual user intervention to locally refine (or coarsen) the grid, being this done automatically at runtime. Doing so, the user would just need to specify a reasonable threshold for the quality parameter driving the AMR algorithm in the setup phase.
- Implementation of an eddy-viscosity two-equations RANS turbulence model with curvature correction for virtual test bench applications. Sensitization to streamline curvature of an eddy-viscosity RANS model could be implemented accordingly to the work done by Smirnov in [82]. This could provide predictive improvements similar to the ones obtained with RSTM RANS models but with significantly reduced computational costs.

6.4 Scientific production

Contextual to this Doctoral thesis project, the following peer-reviewed articles have been produced:

- *“Numerical Evaluation of the Applicability of Steady Test Bench Swirl Ratios to Diesel Engine Dynamic Conditions”*

C. Forte, C. Catellani, G. Cazzoli, G. M. Bianchi, S. Falfari, F. Brusiani, A. Verzè, and S. Saracino. Energy Procedia, 2015. 69th Conference of the Italian Thermal Engineering Association, ATI 2014.

- *“Assessment of Advanced SGS Models for LES Analysis of ICE Wall-Bounded Flows - Part I: Basic Test Case”*

C. Catellani, G. M. Bianchi, S. Falfari, C. Forte, G. Cazzoli. SAE - International Journal of Engines - V125-3, 2016.

Appendices

APPENDIX *A*

Implemented SGS models source code

The present appendix provides the source code related to the dynamic WALE and Sigma SGS models discussed in Section 2.4. The incompressible formulation is here reported.

A.0.1 Dynamic WALE SGS model

- **dynWale.C**

```
1  /*-----*/
2  ===== |
3  \ \      / F i e l d      | OpenFOAM: The Open Source CFD Toolbox
4  \ \      / O p e r a t i o n |
5  \ \      / A n d           | Copyright (C) 1991–2010 OpenCFD Ltd.
6  \ \      / M a n i p u l a t i o n |
7  ----- |
8  License
9      This file is part of OpenFOAM.
10
11      OpenFOAM is free software: you can redistribute it and/or
12      modify it under the terms of the GNU General Public License as
13      published by the Free Software Foundation, either version 3 of
14      the License, or (at your option) any later version.
15
16      OpenFOAM is distributed in the hope that it will be useful,
17      but WITHOUT ANY WARRANTY; without even the implied warranty of
18      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
```

Appendix A. Implemented SGS models source code

```
19     GNU General Public License for more details.
20
21     You should have received a copy of the GNU General Public
22     License along with OpenFOAM.
23     If not, see <http://www.gnu.org/licenses/>.
24
25 Author
26     Cristian Catellani
27     DIN – University of Bologna
28
29 Incompressible dynamic wale model
30     Implementation according to original H. Baya Toda paper
31     "Is the dynamic procedure appropriate for all SGS models?"
32     ECCOMAS CFD 2010, H. Baya Toda, K. Truffin, F. Nicoud
33     ~~~~~*/
34
35 #include "dynWale.H"
36 #include "addToRunTimeSelectionTable.H"
37
38 // * * * * *
39
40 namespace Foam
41 {
42     namespace incompressible
43     {
44         namespace LESModels
45         {
46
47             // * * * * * Static Data Members * * * * *
48
49             defineTypeNameAndDebug(dynWale, 0);
50             addToRunTimeSelectionTable(LESModel, dynWale, dictionary);
51
52             // * * * * * Private Member Functions * * * * *
53
54             void dynWale::updateSubGridScaleFields(const volSymmTensorField& D)
55             {
56                 nuSgs_.correctBoundaryConditions();
57             }
58
59             // * * * * * Constructors * * * * *
60
61             dynWale::dynWale
62             (
63                 const volVectorField& U,
64                 const surfaceScalarField& phi,
65                 transportModel& transport,
66                 const word& turbulenceModelName,
67                 const word& modelName
68             )
69             :
70                 LESModel(modelName, U, phi, transport, turbulenceModelName),
71                 GenEddyVisc(U, phi, transport),
72
73                 k_
74                 (
75                     IOobject
76                     (
```

```

77         "k",
78         runTime_.timeName() ,
79         mesh_,
80         IOobject::NO_READ,
81         IOobject::AUTO_WRITE
82     ),
83     mesh_,
84     dimensionedScalar("k", dimensionSet(0,2,-2,0,0,0), SMALL)
85 ),
86
87 epsilon_
88 (
89     IOobject
90     (
91         "epsilon",
92         runTime_.timeName() ,
93         mesh_,
94         IOobject::NO_READ,
95         IOobject::AUTO_WRITE
96     ),
97     mesh_,
98     dimensionedScalar("epsilon", dimensionSet(0,2,-3,0,0,0),
99                         SMALL)
100 ),
101 epsilonMin_("epsilonMin", kMin_.dimensions()/dimTime, SMALL),
102
103 SVS_
104 (
105     IOobject
106     (
107         "SVS",
108         runTime_.timeName() ,
109         mesh_,
110         IOobject::NO_READ,
111         IOobject::AUTO_WRITE
112     ),
113     mesh_,
114     dimensionedScalar("SVS", dimless, SMALL)
115 ),
116
117 cwsvs2_
118 (
119     IOobject
120     (
121         "cwsvs2",
122         runTime_.timeName() ,
123         mesh_,
124         IOobject::NO_READ,
125         IOobject::AUTO_WRITE
126     ),
127     mesh_,
128     dimensionedScalar("cwsvs2", dimless, SMALL)
129 ),
130
131
132 filterPtr_(LESfilter::New(U.mesh(), coeffDict())),
133 filter_(filterPtr_())

```

Appendix A. Implemented SGS models source code

```
134
135 {
136     bound(k_, kMin_);
137     bound(epsilon_, epsilonMin_);
138
139     updateSubGridScaleFields (dev (symm ( fvc :: grad (U) )));
140     printCoeffs ();
141 }
142
143 // * * * * * Member Functions * * * * * //
144 void dynWale::correct(const tmp<volTensorField>& tgradU)
145 {
146     const volTensorField& gradU = tgradU();
147
148     LESModel::correct(gradU);
149
150     // Compute WALE quantities
151     volTensorField G = gradU & gradU;
152     volSymmTensorField S = symm(gradU);
153
154     volScalarField SS = magSqr(S);
155     volTensorField Sd = dev(G) - skew(G);
156     volScalarField SdSd = magSqr(Sd);
157
158     const volSymmTensorField D(dev(S));
159
160     volScalarField OP1 = SdSd*sqrt(SdSd);
161     volScalarField OP2 = sqr(SS)*sqrt(SS) + SdSd*sqrt(sqrt(SdSd));
162     volScalarField OP = OP1 / OP2;
163
164     // Compute sensor SVS (shear and vortex sensor)
165     Info << "Computing SVS\n";
166     volScalarField SVS1 = SdSd*sqrt(SdSd);
167     volScalarField SVS2 = SdSd*sqrt(SdSd) + sqr(SS)*SS;
168     SVS_ = SVS1 / SVS2;
169
170     // Compute the coefficient cwsvs2 depending on SVS value
171     // accordingly to eqn 25 and 26 of paper
172
173     // Compute Lij and Mij^W as in eqns 9–12 of paper
174     Info << "Computing Mw\n";
175     volSymmTensorField Mw = sqr(delta())*(filter_(OP*(D)) - 4*mag(
176         filter_(OP))*filter_(D));
177     Info << "Computing MwMw\n";
178     dimensionedScalar MwMw = average(magSqr(Mw));
179     if (MwMw.value() > VSMALL)
180     {
181         tmp<volSymmTensorField> LL =
182             dev(filter_(sqr(U)) - (sqr(filter_(U)))));
183
184         cwsvs2_ = 0.5*average(LL && Mw)/MwMw;
185     }
186     else
187     {
188         // Clipping nuEff = nuSgs + nuLam
189         // if nuSgs = 0 -> nuEff = nuLam
190         cwsvs2_ = 0.0;
```

```

191         }
192
193         // Recompute the cwsvs2_ coeff if SVS < 0.09 according
194         // to eqn 25 and 26 of paper
195         forAll(mesh_.C(), cellI)
196         {
197             if (SVS_[cellI] < 0.09)
198             {
199                 cwsvs2_[cellI] = 0.25;
200             }
201         }
202         // Compute nuSgs
203         nuSgs_ = cwsvs2_ * sqr(delta()) * OP;
204         nuSgs_.correctBoundaryConditions();
205
206         // Compute ksgs
207         const scalar cI = 0.101321;
208         k_ = 2.0 * cI * sqr(delta()) * magSqr(D);
209         bound(k_, kMin_);
210
211         // Compute epsilonSgs
212         epsilon_ = 2.0 * nuSgs_ * (magSqr(D));
213         bound(epsilon_, epsilonMin_);
214
215         updateSubGridScaleFields(D);
216
217     }
218
219     bool dynWale::read()
220     {
221         if (GenEddyVisc::read())
222         {
223             filter_.read(coeffDict());
224
225             return true;
226         }
227         else
228         {
229             return false;
230         }
231     }
232
233     // * * * * *
234
235 } // End namespace LESModels
236 } // End namespace incompressible
237 } // End namespace Foam
238
239 // ***** //
```

• dynWale.H

```
1  /*-----*/
2  ===== |
3  \ \      /  F i e l d      | OpenFOAM: The Open Source CFD Toolbox
4  \ \      /  O peration    |
5  \ \      /  A nd          | Copyright (C) 1991–2010 OpenCFD Ltd.
6  \ \      /  M anipulation  |
7  -----*/
8  License
9      This file is part of OpenFOAM.
10
11      OpenFOAM is free software: you can redistribute it and/or
12      modify it under the terms of the GNU General Public License as
13      published by the Free Software Foundation, either version 3 of
14      the License, or (at your option) any later version.
15
16      OpenFOAM is distributed in the hope that it will be useful,
17      but WITHOUT ANY WARRANTY; without even the implied warranty of
18      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
19      GNU General Public License for more details.
20
21      You should have received a copy of the GNU General Public
22      License along with OpenFOAM.
23      If not, see <http://www.gnu.org/licenses/>.
24
25  Author
26      Cristian Catellani
27      DIN – University of Bologna
28
29  Class
30      Foam::incompressible::LESModels::dynWale
31
32  Description
33      Incompressible Wale SGS Model for incompressible flows
34      Eddy viscosity SGS model with k and epsilon sgs output
35
36  SourceFiles
37      dynWale.C
38
39  /*-----*/
40
41  #ifndef dynWale_H
42  #define dynWale_H
43
44  //#include "GenEddyVisc.H"
45  #include "Smagorinsky.H"
46  #include "LESfilter.H"
47
48  // * * * * *
49
50  namespace Foam
51  {
52      namespace incompressible
53      {
54          namespace LESModels
55          {
56
57  /*-----*/
```

```

58                                     Class dynWale Declaration
59  /*-----*/
60
61  class dynWale
62  :
63      public GenEddyVisc
64  {
65      // Private data
66
67          volScalarField k_;
68          volScalarField epsilon_;
69          dimensionedScalar epsilonMin_;
70          volScalarField SVS_;
71          volScalarField cwsvs2_;
72
73          autoPtr<LESfilter> filterPtr_;
74          LESfilter& filter_;
75
76      // Private Member Functions
77
78          //-- Update sub-grid scale fields
79          void updateSubGridScaleFields(const volSymmTensorField& D);
80          //void correct();
81
82          // Disallow default bitwise copy construct and assignment
83          dynWale(const dynWale&);
84          dynWale& operator=(const dynWale&);
85
86  public:
87
88      //-- Runtime type information
89      TypeName("dynWale");
90
91      // Constructors
92
93      //-- Constructor from components
94      dynWale
95      (
96          const volVectorField& U,
97          const surfaceScalarField& phi,
98          transportModel& transport,
99          const word& turbulenceModelName = turbulenceModel::
            typeName,
100         const word& modelName = typeName
101     );
102
103     // Destructor
104     virtual ~dynWale()
105     {}
106
107     // Member Functions
108
109         //-- Return SGS kinetic energy
110         virtual tmp<volScalarField> k() const
111         {
112             return k_;
113         }
114

```

Appendix A. Implemented SGS models source code

```
115         virtual tmp<volScalarField> epsilon() const
116         {
117             return epsilon_;
118         }
119
120         virtual tmp<volScalarField> SVS() const
121         {
122             return SVS_;
123         }
124
125         virtual tmp<volScalarField> cwsvs() const
126         {
127             return cwsvs2_;
128         }
129
130         //– Correct Eddy–Viscosity and related properties
131         virtual void correct(const tmp<volTensorField>& gradU);
132
133         //– Read LESProperties dictionary
134         virtual bool read();
135     };
136
137     // * * * * *
138
139 } // End namespace LESModels
140 } // End namespace incompressible
141 } // End namespace Foam
142
143 // * * * * *
144
145 #endif
146
147 // ***** //
```

A.0.2 Sigma SGS model

• sigma.C

```
1  /*-----*
2  \\      / F i e l d      |
3  \\      / O p e r a t i o n | OpenFOAM: The Open Source CFD Toolbox
4  \\      / A n d      | Copyright (C) 1991–2010 OpenCFD Ltd.
5  \\      / M a n i p u l a t i o n |
6  \\
7  -----*
8  License
9      This file is part of OpenFOAM.
10
11      OpenFOAM is free software: you can redistribute it and/or
12      modify it under the terms of the GNU General Public License as
13      published by the Free Software Foundation, either version 3 of
14      the License, or (at your option) any later version.
15
16      OpenFOAM is distributed in the hope that it will be useful,
17      but WITHOUT ANY WARRANTY; without even the implied warranty of
18      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
19      GNU General Public License for more details.
20
21      You should have received a copy of the GNU General Public
22      License along with OpenFOAM.
23      If not, see <http://www.gnu.org/licenses/>.
24
25  Author
26      Cristian Catellani
27      DIN – University of Bologna
28
29  Incompressible Wale model
30      Implementation according to original H. Baya Toda paper:
31      "A subgrid-scale model based on singular values fo LES in
32      complex geometries"
33
34  /*-----*/
35
36  #include "sigma.H"
37  #include "addToRunTimeSelectionTable.H"
38
39  // *****
40
41  namespace Foam
42  {
43      namespace incompressible
44      {
45          namespace LESModels
46          {
47
48              // ***** Static Data Members ***** //
49
50              defineTypeNameAndDebug(sigma, 0);
51              addToRunTimeSelectionTable(LESModel, sigma, dictionary);
52
53              // ***** Private Member Functions ***** //
54
55              void sigma::updateSubGridScaleFields(const volSymmTensorField& D)
```

Appendix A. Implemented SGS models source code

```
56 {
57     nuSgs_.correctBoundaryConditions();
58 }
59
60 // * * * * * Constructors * * * * * //
61
62 sigma::sigma
63 (
64     const volVectorField& U,
65     const surfaceScalarField& phi,
66     transportModel& transport,
67     const word& turbulenceModelName,
68     const word& modelName
69 )
70 :
71     LESModel(modelName, U, phi, transport, turbulenceModelName),
72     GenEddyVisc(U, phi, transport),
73
74     k_
75     (
76         IOobject
77         (
78             "k",
79             runTime_.timeName(),
80             mesh_,
81             IOobject::NO_READ,
82             IOobject::AUTO_WRITE
83         ),
84         mesh_,
85         dimensionedScalar("k", dimensionSet(0,2,-2,0,0,0,0), SMALL)
86     ),
87
88     epsilon_
89     (
90         IOobject
91         (
92             "epsilon",
93             runTime_.timeName(),
94             mesh_,
95             IOobject::NO_READ,
96             IOobject::AUTO_WRITE
97         ),
98         mesh_,
99         dimensionedScalar("epsilon", dimensionSet(0,2,-3,0,0,0,0),
100             SMALL)
101     ),
102     epsilonMin_("epsilonMin", kMin_.dimensions()/dimTime, SMALL),
103
104     // Sigma specific quantities
105     signal_
106     (
107         IOobject
108         (
109             "signal",
110             runTime_.timeName(),
111             mesh_,
112             IOobject::NO_READ,
```

```

113         IOobject::AUTO_WRITE
114     ),
115     mesh_,
116     dimensionedScalar("sigma1", dimensionSet(0,0,-1,0,0,0,0),
        SMALL)
117 ),
118
119 sigma2_
120 (
121     IOobject
122     (
123         "sigma2",
124         runTime_.timeName(),
125         mesh_,
126         IOobject::NO_READ,
127         IOobject::AUTO_WRITE
128     ),
129     mesh_,
130     dimensionedScalar("sigma2", dimensionSet(0,0,-1,0,0,0,0),
        SMALL)
131 ),
132
133 sigma3_
134 (
135     IOobject
136     (
137         "sigma3",
138         runTime_.timeName(),
139         mesh_,
140         IOobject::NO_READ,
141         IOobject::AUTO_WRITE
142     ),
143     mesh_,
144     dimensionedScalar("sigma3", dimensionSet(0,0,-1,0,0,0,0),
        SMALL)
145 ),
146
147 csigma_
148 (
149     dimensioned<scalar>::lookupOrAddToDict
150     (
151         "csigma",
152         coeffDict_,
153         1.5
154     )
155 ),
156
157 filterPtr_(LESfilter::New(U.mesh(), coeffDict())),
158 filter_(filterPtr_())
159
160 {
161     bound(k_, kMin_);
162     bound(epsilon_, epsilonMin_);
163
164     updateSubGridScaleFields(dev(symm(fvc::grad(U))));
165     printCoeffs();
166 }
167

```

Appendix A. Implemented SGS models source code

```
168 // * * * * * Member Functions * * * * * //
169
170 void sigma::correct(const tmp<volTensorField>& tgradU)
171 {
172     const volTensorField& gradU = tgradU();
173
174     LESModel::correct(gradU);
175
176     // Compute sigma components
177     volTensorField g = gradU;
178     volSymmTensorField G = symm( g.T() & g );
179
180     // Compute the singular values of G
181     // i.e. sigma1, sigma2 and sigma3
182     Info << "Computing sigmas\n";
183     forAll(mesh_.C(), cellI)
184     {
185         // Take a reference G tensor in the current cell
186         const dimensionedSymmTensor& GCurrent = G[cellI];
187         // Evaluate its eigenvalues
188         dimensionedVector Geigen = eigenValues(GCurrent);
189         // The routine returns the eigenvalues sorting them in
190         // increasing order of magnitude
191         sigma3_[cellI] = sqrt( max(SMALL, Geigen.value().x()) );
192         sigma2_[cellI] = sqrt( max(SMALL, Geigen.value().y()) );
193         sigma1_[cellI] = sqrt( max(SMALL, Geigen.value().z()) );
194     }
195
196     // Compute the differential operator Dsigma
197     Info << "Computing Dsigma\n";
198     volScalarField Dsigma = (sigma3_ * (sigma1_ - sigma2_) * \
199                             (sigma2_ - sigma3_) / (sigma1_ * \
200                             sigma1_ + sigma1_*sigma1_*SMALL));
201
202     // Compute nuSgs
203     Info << "Computing nuSgs\n";
204     nuSgs_ = (sqr(csigma_*delta())) * Dsigma;
205     nuSgs_.correctBoundaryConditions();
206
207     // Compute ksgs
208     const volSymmTensorField D(dev(symm(gradU)));
209     const scalar cI = 0.101321;
210     k_ = 2.0 * cI * sqr(delta()) * magSqr(D);
211     bound(k_, kMin_);
212
213     // Compute epsilonSgs
214     epsilon_ = 2.0 * nuSgs_ * (magSqr(D));
215     bound(epsilon_, epsilonMin_);
216
217     updateSubGridScaleFields(D);
218
219 }
220
221 bool sigma::read()
222 {
223     if (GenEddyVisc::read())
224     {
225         csigma_.readIfPresent(coeffDict());
226     }
227 }
```

```
226
227         return true;
228     }
229     else
230     {
231         return false;
232     }
233 }
234
235 // * * * * *
236
237 } // End namespace LESModels
238 } // End namespace incompressible
239 } // End namespace Foam
240
241 // ***** //
```

• sigma.H

```
1  /*-----*
2  ===== |
3  \ \      /  F i e l d      | OpenFOAM: The Open Source CFD Toolbox
4  \ \      /  O p e r a t i o n      |
5  \ \      /  A n d      | Copyright (C) 1991–2010 OpenCFD Ltd.
6  \ \      /  M a n i p u l a t i o n      |
7  -----*
8  License
9      This file is part of OpenFOAM.
10
11      OpenFOAM is free software: you can redistribute it and/or
12      modify it under the terms of the GNU General Public License as
13      published by the Free Software Foundation, either version 3 of
14      the License, or (at your option) any later version.
15
16      OpenFOAM is distributed in the hope that it will be useful,
17      but WITHOUT ANY WARRANTY; without even the implied warranty of
18      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
19      GNU General Public License for more details.
20
21      You should have received a copy of the GNU General Public
22      License along with OpenFOAM.
23      If not, see <http://www.gnu.org/licenses/>.
24
25  Author
26      Cristian Catellani
27      DIN – University of Bologna
28
29  Class
30      Foam::incompressible::LESModels::sigma
31
32  Description
33      Incompressible Sigma SGS Model for incompressible flows
34      Eddy viscosity SGS model with k and epsilon sgs output
35
36  SourceFiles
37      sigma.C
38
39  /*-----*/
40
41  #ifndef sigma_H
42  #define sigma_H
43
44  //#include "GenEddyVisc.H"
45  #include "Smagorinsky.H"
46  #include "LESfilter.H"
47
48  // * * * * *
49
50  namespace Foam
51  {
52      namespace incompressible
53      {
54          namespace LESModels
55          {
56
57  /*-----*
58  ===== |
59  \ \      /  F i e l d      | OpenFOAM: The Open Source CFD Toolbox
60  \ \      /  O p e r a t i o n      |
61  \ \      /  A n d      | Copyright (C) 1991–2010 OpenCFD Ltd.
62  \ \      /  M a n i p u l a t i o n      |
63  -----*
64  License
65      This file is part of OpenFOAM.
66
67      OpenFOAM is free software: you can redistribute it and/or
68      modify it under the terms of the GNU General Public License as
69      published by the Free Software Foundation, either version 3 of
70      the License, or (at your option) any later version.
71
72      OpenFOAM is distributed in the hope that it will be useful,
73      but WITHOUT ANY WARRANTY; without even the implied warranty of
74      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
75      GNU General Public License for more details.
76
77      You should have received a copy of the GNU General Public
78      License along with OpenFOAM.
79      If not, see <http://www.gnu.org/licenses/>.
80
81  Author
82      Cristian Catellani
83      DIN – University of Bologna
84
85  Class
86      Foam::incompressible::LESModels::sigma
87
88  Description
89      Incompressible Sigma SGS Model for incompressible flows
90      Eddy viscosity SGS model with k and epsilon sgs output
91
92  SourceFiles
93      sigma.C
94
95  /*-----*/
96
97  #ifndef sigma_H
98  #define sigma_H
99
100  //#include "GenEddyVisc.H"
101  #include "Smagorinsky.H"
102  #include "LESfilter.H"
103
104  // * * * * *
105
106  namespace Foam
107  {
108      namespace incompressible
109      {
110          namespace LESModels
111          {
112
```

```

58                                     Class sigma Declaration
59  *-----*/
60
61  class sigma
62  :
63      public GenEddyVisc
64  {
65      // Private data
66
67          volScalarField k_;
68          volScalarField epsilon_;
69          dimensionedScalar epsilonMin_;
70
71          // Specific of sigma SGS model
72          volScalarField sigma1_;
73          volScalarField sigma2_;
74          volScalarField sigma3_;
75          dimensionedScalar csigma_;
76
77          autoPtr<LESfilter> filterPtr_;
78          LESfilter& filter_;
79
80      // Private Member Functions
81
82          //-- Update sub-grid scale fields
83          void updateSubGridScaleFields(const volSymmTensorField& D);
84          //void correct();
85
86          // Disallow default bitwise copy construct and assignment
87          sigma(const sigma&);
88          sigma& operator=(const sigma&);
89
90  public:
91
92      //-- Runtime type information
93      TypeName("sigma");
94
95      // Constructors
96
97      //-- Constructor from components
98      sigma
99      (
100          const volVectorField& U,
101          const surfaceScalarField& phi,
102          transportModel& transport,
103          const word& turbulenceModelName = turbulenceModel::
              typeName,
104          const word& modelName = typeName
105      );
106
107      // Destructor
108      virtual ~sigma()
109      {}
110
111      // Member Functions
112
113          //-- Return SGS kinetic energy
114          virtual tmp<volScalarField> k() const

```

Appendix A. Implemented SGS models source code

```
115         {
116             return k_;
117         }
118
119         virtual tmp<volScalarField> epsilon() const
120         {
121             return epsilon_;
122         }
123
124         //-- Correct Eddy-Viscosity and related properties
125         virtual void correct(const tmp<volTensorField>& gradU);
126
127         //-- Read LESProperties dictionary
128         virtual bool read();
129     };
130
131     // * * * * *
132
133 } // End namespace LESModels
134 } // End namespace incompressible
135 } // End namespace Foam
136
137 // * * * * *
138
139 #endif
140
141 // *****
```

APPENDIX *B*

Mass Flow Rate and Swirl Torque evaluation methodology

B.0.3 Mass Flow Rate and Swirl Torque evaluation

The effect of the swirl meter on the flow developing inside the cylinder is basically the blockage of the velocity components normal to the cylinder axis, as explained in Paragraph 5.2.2, accompanied by a minimal pressure drop. In the same section, the computational modeling of the swirl meter as a porous region is also explained.

The evaluation of the desired quantities is made on a section plane perpendicular to the cylinder axis and placed at least one cell far from the beginning of the porous region, as shown in Figure B.1. Such distance has to be observed in order to avoid potential numerical disturbances of the momentum sources active in the porous region onto the measurement plane.

The flow straightening effect given by the swirl meter is the result of the angular momentum exchange between the fluid and the instrument. The swirl torque can be then computed evaluating the variation of the (vector) angular momentum \mathbf{L} across the porous region that models the swirl meter:

$$\Delta \mathbf{L} = \mathbf{L}_{ups} - \mathbf{L}_{dws} \quad (\text{B.1})$$

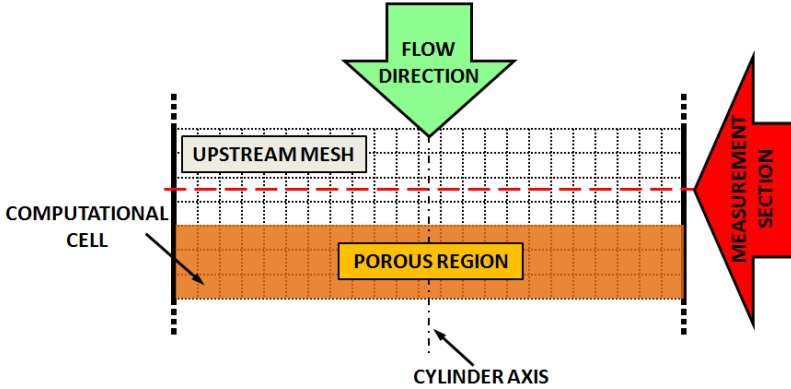


Figure B.1: Swirl torque measurement section location.

where ΔL is the variation of the fluid angular momentum, evaluated with respect to the cylinder axis, between the swirl meter inlet section (subscript *ups*) and outlet section (subscript *dws*).

It has to be noted that every considered angular momentum has to be computed with respect to the cylinder axis, since that is also the axis about which the swirl meter operates and the swirl torque is retrieved.

Due to the flow straightening operated by the instrument, the residual swirl at the instrument outlet section is negligible (as also visualized by the streamlines of Figure 5.10), and so is the angular momentum. Expression (B.1) then simplifies as:

$$\Delta L \approx L_{ups} \quad (B.2)$$

meaning that the momentum variation can be computed just by evaluation of the section upstream the instrument.

OpenFOAM[®] post-processing tools allow to extract the desired section plane from the full 3D simulation results. The section is performed by slicing the cells at the level of the plane and triangulating the resulting topology. Finally, the computed fields of the 3D mesh are interpolated onto the 2D mesh of the circular slice so obtained.

An exemplary 2D section is reported in Figure B.2. Given a generic cell *i*, it is possible to decompose the velocity U_i in two (vector) components:

- U_{ax} , the axial velocity component,
- U_{nor} , the velocity component normal to the cylinder axis.

Then, the angular momentum contribution of cell *i* with respect to the cylinder axis is given by:

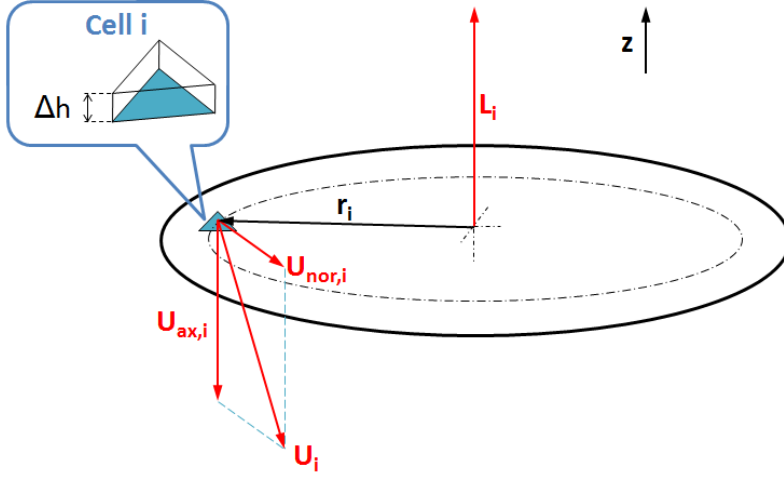


Figure B.2: Swirl torque computation.

$$\mathbf{L}_i = \mathbf{r}_i \times (m_i \mathbf{U}_{nor,i}) \quad (\text{B.3})$$

where \mathbf{r}_i is the distance vector between the cell i center and the cylinder axis and m_i is the mass ascribed to the cell. Such mass can be evaluated by imagining the section cells having thickness Δh , as shown in Figure B.2. This leads to the following expression for the computation of m_i :

$$m_i = \rho_i V_i = \rho_i A_i \Delta h \quad (\text{B.4})$$

Plugging Expression (B.4) into Equation (B.3), the angular momentum contribution reads:

$$\mathbf{L}_i = \mathbf{r}_i \times (\rho_i A_i \Delta h \mathbf{U}_{nor,i}) \quad (\text{B.5})$$

and the total angular momentum is then the sum of all the cells individual contributions:

$$\mathbf{L} = \sum_i \mathbf{L}_i \quad (\text{B.6})$$

From *Newton's Second law of Motion* it is then possible to compute the swirl torque as:

$$\mathbf{T} = \frac{\Delta \mathbf{L}}{\Delta t} \approx \frac{\mathbf{L}_{ups}}{\Delta t} = \sum_i^{ups} \frac{\mathbf{r}_i \times (\rho_i A_i \Delta h \mathbf{U}_{nor,i})}{\Delta t} \quad (\text{B.7})$$

The ratio $\Delta h / \Delta t$ corresponds to the average axial velocity of the fluid in cell i . If $\Delta h \rightarrow 0$, then, such ratio tends to the axial velocity $U_{ax,i}$, being $U_{ax,i} = |U_{ax,i}|$. Equation (B.7) can be therefore rearranged as:

$$\mathbf{T} \approx \sum_i^{ups} \mathbf{r}_i \times (\rho_i A_i |U_{ax,i}| U_{nor,i}) \quad (\text{B.8})$$

It must be noted, however, that local recirculation of fluid can occur on the sampling plane. If $U_{ax,i}$ has positive z direction, it means that the fluid parcel related to cell i is moving upstream, against the main flow direction. In such case, the fluid parcel is not entering the porous region and is not going to interact with the swirl meter. It is therefore important to remove its contribution from summation (B.8).

This is done by introducing the following backflow detector parameter:

$$bk_i = \begin{cases} 0 & \text{for } U_{ax,i} \cdot \mathbf{n}_z \geq 0 \\ 1 & \text{for } U_{ax,i} \cdot \mathbf{n}_z < 0 \end{cases} \quad (\text{B.9})$$

where \mathbf{n}_z is the normal unit z vector. The bk parameter is then plugged into Equation (B.8) to exploit its function:

$$\mathbf{T} \approx \sum_i^{ups} \mathbf{r}_i \times (bk_i \rho_i A_i |U_{ax,i}| U_{nor,i}) \quad (\text{B.10})$$

The torque \mathbf{T} here obtained is a vector quantity. In the present work its modulus has been used since its direction is always coincident with the main swirling vortex direction, i.e. the engine head swirl direction of design.

Finally, the mass flow rate of the system can be computed using the same post-processing topology used for the swirl torque evaluation. More precisely, the overall mass flow rate at the inlet section of the swirl meter can be computed as a summation of all the individual cells mass flow rates:

$$mfr = \sum_i^{ups} \rho_i A_i |U_{ax,i}| (-\text{sign}(U_{ax,i} \cdot \mathbf{n}_z)) \quad (\text{B.11})$$

where, thanks to the last term, the contribution of the backflowing fluid is correctly accounted for, allowing to retrieve the net mass flow rate through the section.

B.0.4 Post-processing script

The methodology described in the previous paragraph for mass flow rate and swirl torque evaluation has been implemented in a Python script, here reported:

```
1 #####
2 #####
3 #####          TORQUE AND MFR COMPUTING SCRIPT          #####
4 #####
5 #####
6
7 # Python modules import.
8 import sys
9 import numpy as np
10
11 __author__ = "Cristian Catellani"
12 __copyright__ = "Copyright 2016 – Cristian Catellani"
13 __credits__ = ["Cristian Catellani", "Giulio Cazzoli"]
14 __license__ = "GPL"
15 __version__ = "0.1"
16 __maintainer__ = "Cristian Catellani"
17 __email__ = "cristian.catellani5@unibo.it"
18 __status__ = "Development"
19
20 #####
21 #####          USER INPUT SECTION          #####
22 #####
23 # Data files names.
24 rhoFile='rhoMean_coord.dat'
25 UFile='UMean_area.dat'
26
27 # Slice normal, i.e. cylinder axis direction.
28 # Please note that it has to point towards main flow direction.
29 dirAxis=np.array([0,0,-1])
30
31 #####
32 #####          DATA READ-IN          #####
33 #####
34 # Read-in the data.
35 rhoFileData=[]
36 UFileData=[]
37
38 with open(rhoFile,'r') as D1:
39     for line in D1:
40         rhoFileData.append([float(x) for x in line.split()])
41
42 with open(UFile,'r') as D2:
43     for line in D2:
44         UFileData.append([float(x) for x in line.split()])
45
46 #####
47 #####          NUMPY DATA STRUCTURES GENERATION          #####
48 #####
49 # Numpyfy data.
50 rhoFileDataNP=np.array(rhoFileData)
51 UFileDataNP=np.array(UFileData)
52
```

Appendix B. Mass Flow Rate and Swirl Torque evaluation methodology

```
53 # Retrieve single quantities.
54 ccx=rhoFileDataNP[:,1]           # cells coordinate 'x'
55 ccy=rhoFileDataNP[:,2]           # cells coordinate 'y'
56 ccz=rhoFileDataNP[:,3]           # cells coordinate 'z'
57 rho=rhoFileDataNP[:,0]           # cells density
58 U_x=UFileDataNP[:,0]             # cells 'x' vel component
59 U_y=UFileDataNP[:,1]             # cells 'y' vel component
60 U_z=UFileDataNP[:,2]             # cells 'z' vel component
61 area=UFileDataNP[:,3]            # cells area
62
63 #####
64 #####          COMPUTE RADII AND VELOCITIES          #####
65 #####
66 # Compute slice center (the slice could be not perfectly centered...)
67 xCenter=np.mean(ccx)
68 yCenter=np.mean(ccy)
69 zCenter=np.mean(ccz)
70
71 # Initialize radius and velocity vector fo each cell.
72 nCells=np.size(ccx)
73 radius=np.empty([nCells,3]); radius.fill(9999)
74 velocity=np.empty([nCells,3]); velocity.fill(12345)
75
76 # Compute cells radii.
77 radius[:,0]=ccx[:]-xCenter
78 radius[:,1]=ccy[:]-yCenter
79 radius[:,2]=ccz[:]-zCenter
80
81 # Compute cells velocities.
82 velocity[:,0]=U_x[: ]
83 velocity[:,1]=U_y[: ]
84 velocity[:,2]=U_z[: ]
85
86 #####
87 #####          COMPUTE MASS FLOW RATE          #####
88 #####
89 # Compute velocity normal to the slice, i.e. velocity component
90 # parallel to cyl axis.
91 UaxComp=np.dot(velocity,dirAxis)
92
93 # Compute mfr of single cells
94 mfrSingleCell=rho*UaxComp*area
95
96 # Overall mfr
97 mfr=np.sum(mfrSingleCell)
98
99 #####
100 #####          COMPUTE SWIRL TORQUE          #####
101 #####
102 # Compute velocity components parallel and normal to cyl axis (Uax, Unor).
103 unitAxis=dirAxis/np.linalg.norm(dirAxis)
104 Uax=np.empty([nCells,3]); Uax.fill(9999)
105 for jj in range(nCells):
106     Uax[jj]=np.multiply(UaxComp[jj],unitAxis)
107 Unor=velocity-Uax
108
109 # Check out the direction of axial velocity.
110 # If the fluid is backflowing in front of the swirl meter (i.e. is going
```

```

111 # in the opposite direction respect to the main flow), exclude such
112 # element contribution to torque. This is done multiplying the respective
113 # mfr by 0 through the parameter bk.
114 bk=np.greater(mfrSingleCell,1e-6)
115
116 # Compute cross product between radius and Unor.
117 # Retain just the component parallel to cyl axis, with sign. This is done
118 # through a simple dot product against the axis unit vector. Please note
119 # that the component sign depends on the tangential velocity component
120 # direction of rotation and has nothing to do with the cell's mass flow
121 # direction.
122 crossCells=np.dot( (np.cross(radius,Unor)) , unitAxis )
123
124 # Compute torque
125 torqueCells=crossCells*bk*rho*area*UaxComp
126 torque=np.sum(torqueCells)
127
128 #####
129 #####                                OUTPUT RESULTS                                #####
130 #####
131
132 # Write mfr and torque to file.
133 f = open('results.txt','w')
134 f.write('##### MFR AND TORQUE RESULTS #####\n')
135 f.write('Mass Flow Rate [kg/s]: ' \
136         + str("{:8.6f}".format(mfr)) + '\n')
137 f.write('Swirl toque [Nm]: ' \
138         + str("{:8.6f}".format(torque)) + '\n')
139 f.write('#####\n')
140 f.close()

```

Bibliography

- [1] ACEA. The automobile industry pocket guide 2015-2016, 2015.
- [2] I. I. Wiebe. Semi-empirical expression for combustion rate in engines. In USSR, editor, *Proceedings of Conference on Piston engines*, pages 185–191, 1956.
- [3] G. Woschni. Universally applicable equation for the instantaneous heat transfer coefficient in the internal combustion engine. *SAE*, (670931), 1967.
- [4] A. Amsden, J. Ramshow, P. O'Rourke, and T. Butler. *KIVA: A Computer Program for Two- and Three-Dimensional Fluid Flows with Chemical Reactions and Fuel Sprays*. Los Alamos National Laboratories, 1985.
- [5] N. D. Sandham. A review of progress on direct and large-eddy simulation. In Bernard J. Geurts, editor, *Modern simulation strategies for turbulent flow*, pages 1–20. R.T. Edwards, 2001.
- [6] S. B. Pope. *Turbulent flows*. Cambridge University Press, 2000.
- [7] R. Aris. *Vectors, Tensors and the Basic Equations of Fluid Mechanics*. Dover Publications, 1990.
- [8] J. H. Ferziger and M. Peric. *Computational Methods for Fluid Dynamics*. Springer Berlin Heidelberg, 2001.
- [9] P. Lax and B. Wendroff. Systems of conservation laws. *Communications on Pure and Applied Mathematics*, 13(2):217–237, 1960.
- [10] H. K. Versteeg and W. Malalasekera. *An Introduction to Computational Fluid Dynamics. The Finite Volume Method*. Longman Group Ltd., London, 1995.
- [11] W. Sutherland. Lii. the viscosity of gases and molecular force. *Philosophical Magazine Series* 5, 36(223):507–531, 1893.
- [12] J. Smagorinsky. General circulation experiments with the primitive equations. *Mon. Wea. Rev.*, 91(3):99–164, 1963.
- [13] A. N. Kolmogorov. The local structure of turbulence in incompressible viscous fluid for very large reynolds numbers. volume 30, pages 301–305. JSTOR, 1941.
- [14] P. Sagaut. *Large Eddy Simulation for Incompressible Flows*. Springer Berlin Heidelberg, 2005.

Bibliography

- [15] F. M. Denaro. What does finite volume-based implicit filtering really resolve in large-eddy simulations? *Journal of Computational Physics*, 230(10):3849 – 3883, 2011.
- [16] A. Leonard. Energy cascade in large-eddy simulations of turbulent fluid flows. In F.N. Frenkiel and R.E. Munn, editors, *Turbulent Diffusion in Environmental Pollution Proceedings of a Symposium held at Charlottesville*, volume 18, Part A of *Advances in Geophysics*, pages 237 – 248. Elsevier, 1975.
- [17] J. H. Ferziger. Large eddy numerical simulations of turbulent flows. *AIAA Journal*, 15(9):1261–1267, September 1977.
- [18] D.K. Lilly, NOAA/NCAR Boulder Atmospheric Observatory, and United States. National Center for Atmospheric Research. *On the Application of the Eddy Viscosity Concept in the Inertial Sub-range of Turbulence*. Manuscript. NCAR, 1966.
- [19] E. Garnier, N. Adams, and P. Sagaut. *Large Eddy Simulation for Compressible Flows*. Springer Netherlands, 2009.
- [20] C. J. Rutland. Large-eddy simulations for internal combustion engines - a review. *International Journal of Engine Research*, 12(5):421–451, October 2011.
- [21] N. A. Adams and S. Hickel. *Advances in Turbulence XII: Proceedings of the 12th EUROMECH European Turbulence Conference, September 7-10, 2009, Marburg, Germany*, chapter Implicit Large-Eddy Simulation: Theory and Application, pages 743–750. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.
- [22] F. Grinstein, L. Margolin, and W. Rider. *Implicit Large Eddy Simulation*. Cambridge University Press, 2011.
- [23] M. Germano, U. Piomelli, P. Moin, and W. H. Cabot. A dynamic subgrid scale eddy viscosity model. *Physics of Fluids A: Fluid Dynamics (1989-1993)*, 3(7):1760–1765, 1991.
- [24] D. K. Lilly. A proposed modification of the germano subgrid-scale closure method. *Physics of Fluids*, 1992.
- [25] F. Nicoud and F. Ducros. Subgrid-scale stress modelling based on the square of the velocity gradient tensor. *Flow, Turbulence and Combustion*, 62(3):183–200, 1999.
- [26] H. Baya Toda, K. Truffin, and F. Nicoud. Is the dynamic procedure appropriate for all sgs models? In *Proceedings of ECCOMAS CFD 2010*. ECCOMAS, 2010.
- [27] H. Baya Toda, O. Cabrit, G. Balarac, S. Bose, J. Lee, H. Choi, and F. Nicoud. A subgrid-scale model based on singular values for les in complex geometries. In *Proceedings of Summer Program 2010*. Center for Turbulence Research, 2010.
- [28] A. W. Vreman. An eddy-viscosity subgrid-scale model for turbulent shear flow: Algebraic theory and applications. *Physics of Fluids (1994-present)*, 16(10):3670–3681, 2004.
- [29] J. Crank and P. Nicolson. A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type. *Advances in Computational Mathematics*, 6(1):207–226, 1947.
- [30] C. Hirsch, editor. *Numerical Computation of Internal and External Flows: Fundamentals of Numerical Discretization*. John Wiley & Sons, Inc., New York, NY, USA, 1988.
- [31] J. Blazek. *Computational Fluid Dynamics: Principles and Applications, 3rd Edition*. Elsevier, 2015.
- [32] H. Jasak. *Error Analysis and Estimation for the Finite Volume Method with Applications to Fluid Flows*. PhD thesis, Imperial College of Science, Technology and Medicine - London, 1996.

- [33] H. Jasak, H.G. Weller, and A.D. Gosman. High resolution nvd differencing scheme for arbitrarily unstructured meshes. *International Journal for Numerical Methods in Fluids*, 31(2):431–449, 1999.
- [34] Y. Saad. *Iterative Methods for Sparse Linear Systems, Second Edition*. Society for Industrial and Applied Mathematics, 2003.
- [35] S.V Patankar and D.B Spalding. A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. *International Journal of Heat and Mass Transfer*, 15(10):1787 – 1806, 1972.
- [36] J. P. Van Doormaal and G. D. Raithby. Enhancements of the SIMPLE method for predicting incompressible fluid flows. *Numerical Heat Transfer*, 7(2):147–163, 1984.
- [37] R. I. Issa. Solution of the implicitly discretised fluid flow equations by operator-splitting. *J. Comput. Phys.*, 62(1):40–65, January 1986.
- [38] OpenFOAM Foundation. *OpenFOAM-2.3.0 Programmer’s Guide*, 2014.
- [39] H. G. Weller, G. Tabor, H. Jasak, and C. Fureby. A tensorial approach to computational continuum mechanics using object-oriented techniques. *Comput. Phys.*, 12(6):620–631, 1998.
- [40] H. C. Yee, M. Vinokur, and M. J. Djomehri. Entropy splitting and numerical dissipation. *J. Comput. Phys.*, 162(1):33–81, July 2000.
- [41] P. R. Rao and L. A. Schaefer. Numerical stability of explicit off-lattice boltzmann schemes: A comparative study. *Journal of Computational Physics*, 285(0):251–264, March 2015.
- [42] V. Vuorinen, J.-P. Keskinen, C. Duwig, and B.J. Boersma. On the implementation of low-dissipative runge-kutta projection methods for time dependent flows using openfoam®. *Computers & Fluids*, 93(0):153–163, April 2014.
- [43] V. Moureau, G. Lartigue, Y. Sommerer, C. Angelberger, O. Colin, and T. Poinso. Numerical methods for unsteady compressible multi-component reacting flows on fixed and moving grids. *Journal of Computational Physics*, 202(2):710–736, January 2005.
- [44] F. Brusiani, G. M. Bianchi, T. Baritaud, and A. Bianchi d’Espinosa. Using les for predicting high performance car airbox flow. *SAE Int. J. Passeng. Cars - Mech. Syst.*, 2:1050–1064, 2009.
- [45] G. Ferrari. *Motori a combustione interna*. Il capitello, 2001.
- [46] J. B. Heywood. *Internal Combustion Engine Fundamentals*. Mcgraw-Hill, 1989.
- [47] J. L. Lumley. *Engines: An Introduction*. Cambridge University Press, 1999.
- [48] T. W. Kuo, X. Yang, V. Gopalakrishnan, and Z. Chen. Large eddy simulation (les) for ic engine flows. *Oil Gas Sci. Technol. - Rev. IFP Energies nouvelles*, 69(1):61–81, 2014.
- [49] B. R. Petersen and J. Ghandhi. High resolution scalar dissipation and turbulence length scale measurements in an internal combustion engine. *SAE Int. J. Engines*, 3:65–83, 04 2010.
- [50] D. Heim and J. Ghandhi. A detailed study of in-cylinder flow and turbulence using piv. *SAE Int. J. Engines*, 4:1642–1668, 04 2011.
- [51] F. Brusiani, C. Forte, and G. M. Bianchi. Assessment of a numerical methodology for large eddy simulation of ice wall bounded non-reactive flows. *SAE*, 10 2007.
- [52] F. Piscaglia, A. Montorfano, A. Onorati, and F. Brusiani. Boundary conditions and sgs models for les of wall-bounded separated flows: An application to engine-like geometries. *Oil Gas Sci. Technol. - Rev. IFP Energies nouvelles*, 69(1):11–27, January 2014.
- [53] J. P. Keskinen, V. Vuorinen, O. Kaario, and M. Larmi. Effects of mesh deformation on the quality of large eddy simulations. In *Fluid Dynamics and Co-located Conferences*, pages –. American Institute of Aeronautics and Astronautics, June 2013.

- [54] F. Di Mare, R. Knappstein, and M. Baumann. Application of les-quality criteria to internal combustion engine flows. *Computers & Fluids*, 89(0):200 – 213, 2014.
- [55] A. M. Kempf. Les validation from experiments. *Flow, Turbulence and Combustion*, 80(3):351–373, 2008.
- [56] I. B. Celik, Z. N. Cehreli, and I. Yavuz. Index of resolution quality for large eddy simulations. *Journal of Fluids Engineering*, 127(5):949–, 2005.
- [57] V. K. Krastev, G. Bella, and G. Campitelli. Some developments in des modeling for engine flow simulation. In *SAE Technical Paper*. SAE International, 09 2015.
- [58] S.B. Pope. Ten questions concerning the large-eddy simulation of turbulent flows. *New Journal of Physics*, 6(1):35, 2004.
- [59] P. Durbin and R. B. A. Pettersson. *Statistical theory and modeling for turbulent flows*. Wiley, 2011.
- [60] T. Poinso, M. Garcia, J.M. Senoner, L. Gicquel, G. Staffelbach, and O. Vermorel. Numerical and physical instabilities in massively parallel les of reacting flows. *Journal of Scientific Computing*, 49(1):78–93, 2011.
- [61] B. J. Geurts and J. Fröhlich. A framework for predicting accuracy limitations in large-eddy simulation. *Physics of Fluids (1994-present)*, 14(6):L41–L44, 2002.
- [62] P. A. Dellenback, D. E. Metzger, and G. P. Neitzel. Measurements in turbulent swirling flow through an abrupt axisymmetric expansion. *AIAA Journal*, 26(6):669–681, June 1988.
- [63] J. U. Schlüter, H. Pitsch, and P. Moin. Large-eddy simulation inflow conditions for coupling with reynolds-averaged flow solvers. *AIAA Journal*, 42(3):478–484, March 2004.
- [64] A. Javadi and H. Nilsson. Les and des of strongly swirling turbulent flow through a suddenly expanding circular pipe. *Computers & Fluids*, 107:301–313, 2015.
- [65] F. Piscaglia, A. Montorfano, and A. Onorati. A scale adaptive filtering technique for turbulence modeling of unsteady flows in ic engines. *SAE Int. J. Engines*, 8:426–436, 04 2015.
- [66] W. Gyllenram and H. Nilsson. Design and validation of a scale-adaptive filtering technique for lrn turbulence modeling of unsteady flow. *Journal of Fluid Engineering*, Volume 130:10, 2008.
- [67] L. Thobois, G. Rymer, T. Souleres, and T. Poinso. Large-eddy simulation in ic engine geometries. *SAE Technical Paper*, pages –, 2004.
- [68] F. Brusiani and G. M. Bianchi. Les simulation of ice non-reactive flows in fixed grids. *SAE*, 04 2008.
- [69] J. Martínez, F. Piscaglia, A. Montorfano, A. Onorati, and S.M. Aithal. Influence of spatial discretization schemes on accuracy of explicit les: Canonical problems to engine-like geometries. *Computers & Fluids*, 117:62 – 78, 2015.
- [70] H. Xu. Some critical technical issues on the steady flow testing of cylinder heads. *SAE Technical Paper*, 2001.
- [71] Ricardo. Steady state flowbench port performance measurement and analysis techniques. Technical Report DP93/0704, Ricardo, 1993.
- [72] AVL. Port development - flow test bench. Report collection, AVL List GmbH, 1995.
- [73] Z. Lu, T. Wang, S. Liu, Z. Lin, and Y. Han. Experimental and modeling study of the effect of manufacturing deviations on the flow characteristics of tangential intake port in a diesel engine. *Journal of Engineering for Gas Turbines and Power*, 136(11):112101–112101, May 2014.

-
- [74] C. Forte, C. Catellani, G. Cazzoli, G. M. Bianchi, S. Falfari, F. Brusiani, A. Verzè, and S. Saracino. Numerical evaluation of the applicability of steady test bench swirl ratios to diesel engine dynamic conditions. *Energy Procedia*, 81:732 – 741, 2015. 69th Conference of the Italian Thermal Engineering Association, {ATI} 2014.
- [75] Creative Fields Ltd. *cfMesh v1.1 User's Guide*, 2015.
- [76] V. Yakhot, S. A. Orszag, S. Thangam, T. B. Gatski, and C. G. Speziale. Development of turbulence models for shear flows by a double expansion technique. *Physics of Fluids A*, 4(7):1510–1520, 1992.
- [77] X. Yang, Z. Chen, and T.W. Kuo. Pitfalls for accurate steady-state port flow simulations. *Journal of Engineering for Gas Turbines and Power*, 135(6):061601–061601, May 2013.
- [78] M. F. Palumbo. Cfd-analysis of intake-system performances of a small turbocharged spark-ignition engine. In *SAE Technical Paper*. Consiglio Nazionale delle Ricerche, 09 2007.
- [79] T. Maric, J. Hopken, and K. Mooney. *The OpenFOAM Technology Primer*. Sourceflux, 2014.
- [80] B. E. Launder, G. J. Reece, and W. Rodi. Progress in the development of a reynolds-stress turbulence closure. *Journal of Fluid Mechanics*, 68:537–566, 4 1975.
- [81] C. Meneveau, T. S. Lund, and W. H. Cabot. A lagrangian dynamic subgrid-scale model of turbulence. *Journal of Fluid Mechanics*, 319:353–385, 7 1996.
- [82] P. E. Smirnov and F. R. Menter. Sensitization of the sst turbulence model to rotation and curvature by applying the spalart-shur correction term. *Journal of Turbomachinery*, 131(4):041010–041010, July 2009.